ICS 35.040

L 80

# 中华人民共和国国家标准

GB/T 15852.1—XXXX

代替 GB/T 15852.1-2008

# 信息技术 安全技术 消息鉴别码 第1部分：采用分组密码的机制

Information technology—Security techniques—Message Authentication Codes (MACs)—Part 1: Mechanisms using a block cipher

(ISO/IEC 9797-1:2011,MOD)

（报批稿）

（本稿完成日期：2019年11月5日）

XXXX－XX－XX 发布　　XXXX－XX－XX 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会　发布

# 目　　次

# 前　言

GB/T 15852《信息技术 安全技术 消息鉴别码》由如下部分组成：

——第1部分：采用分组密码的机制；

——第2部分：采用专用杂凑函数的机制；

——第3部分：采用泛杂凑函数的机制。

本部分是GB/T 15852的第1部分。

本部分按照GB/T 1.1-2009《标准化工作导则 第1部分：标准的结构和编写》和GB/T 20000.2《标准化工作指南 第2部分：采用国际标准》给出的规则起草。

本部分代替GB/T 15852.1-2008。

本部分与GB/T 15852.1-2008相比，主要技术变化如下：

——删除了消息鉴别码算法用途的说明（见GB/T 15852.1-2008的第1章）；

——增加了MAC算法的常用名称指代（见引言、第5章、第7章，参见附录A、附录B）；

——增加了规范性引用文件GB/T 32907-2016（见第2章）；

——修改了“术语和定义”的条目顺序（见第3章，GB/T 15852.1-2008的第3章）；

——增加了16个符号，修改了3个符号（见4.1，GB/T 15852.1-2008的第4章）；增加了“缩略语”（见4.2）；

——修改了第5章的标题，将“要求”改为“用户要求”；修改了用户选择密钥诱导方法的要求（见第5章，GB/T 15852.1-2008的第5章）；

——增加了使用MAC算法4时数据串长度的要求；增加了使用MAC算法7时MAC的长度要求（见第5章）；

——修改了MAC算法的一般模型，增加了密钥诱导和最终迭代操作，并修改了“MAC算法模型”图（见6.1，GB/T 15852.1-2008的第6章）；

——增加了密钥诱导操作的概述与方法、最终迭代操作的概述与方法（见6.2、6.7）；增加了填充方法4、初始变换3（见6.3.5、6.5.4）；修改了迭代应用分组密码操作（见6.4）；增加了截断操作的概述和截断操作2（见6.9）；

——修改了MAC算法5，替换为CMAC（见7.6，GB/T 15852.1-2008的7.5）；修改了MAC算法6，替换为LMAC（见7.7，GB/T 15852.1-2008的7.6）；

——增加了MAC算法7（TrCBC）和8（CBCR）（见7.8、7.9）；

——修改了附录“例子”的标题为“测试向量”；修改了使用的分组密码算法，将DEA修改为SM4分组密码算法；修改了明文、密钥、结果（参见附录A，GB/T 15852.1-2008的附录A）；增加了MAC算法7和8的测试向量（参见A.7、A.8）；

——修改了表B.1中序号为1.2和4.2的算法效率；增加了MAC算法7和8的安全性说明、算法的特性、安全强度估计（参见附录B）。

本部分修改采用ISO/IEC 9797-1:2011《信息技术 安全技术 消息鉴别码 第1部分：采用分组密码的机制》。

本部分与国际标准ISO/IEC 9797-1:2011的主要技术差异如下：

——增加了MAC算法的常用名称指代（见引言、第5章、第7章，参见附录A、附录B）；

——删除了密钥管理机制和对象标识符的说明（见ISO/IEC 9797-1:2011的第1章）；

——增加了规范性引用文件 GB/T 9387.2-1995、GB/T 15843.1-2017、GB/T 17964-2008，将 ISO/IEC 18033-3 替换为 GB/T 32907-2016（见第 2 章，ISO/IEC 9797-1:2011 的第 2 章）；

——增加了初始变换等 4 个符号和代号（见 4.1）；增加了“缩略语”（见 4.2）；

——修改了用户选择密钥诱导方法的要求（见第 5 章，ISO/IEC 9797-1:2011 的第 5 章）；

——修改了 MAC 算法的一般模型，增加了初始变换操作，并修改了“MAC 算法模型”图（见 6.1，ISO/IEC 9797-1:2011 的 6.1）；

——增加了填充方法 4、初始变换 3（见 6.3.5、6.5.4）；修改了迭代应用分组密码操作（见 6.4，ISO/IEC 9797-1:2011 的 6.5）；增加了截断操作的概述和截断操作 2（见 6.9）；

——增加了 MAC 算法 7（TrCBC）和 8（CBCR）（见 7.8、7.9）；

——删除了附录“对象标识符”（参见 ISO/IEC 9797-1:2011 的附录 A）；

——修改了附录“例子”的标题为“测试向量”；修改了使用的分组密码算法、明文、密钥、结果（参见附录 A，ISO/IEC 9797-1:2011 的附录 B）；增加了 MAC 算法 7 和 8 的测试向量（参见 A.7、A.8）；

——修改了表 B.1 中序号为 1.2 和 4.2 的算法效率（参见附录 B，ISO/IEC 9797-1:2011 的附录 C）；增加了 MAC 算法 7 和 8 的安全性说明、算法的特性、安全强度估计（参见附录 B）；

——删除了获得高安全性强度的 MAC 算法的方法及建议（参见 ISO/IEC 9797-1:2011 的 C.2）；

——删除了附录“与以前的 MAC 算法标准的比较”（参见 ISO/IEC 9797-1:2011 的附录 D）。

请注意本部分的某些内容可能涉及专利，本部分的发布机构不承担识别这些专利的责任。

本部分由全国信息安全标准化技术委员会（SAC/TC260）提出并归口。

本部分起草单位：中国科学院软件研究所、成都卫士通信息产业股份有限公司、桂林电子科技大学、国家密码管理局商用密码检测中心。

本部分主要起草人：吴文玲、眭晗、张立廷、张蕾、韦永壮、毛颖颖、郑雅菲、涂彬彬、刘仁章、丁勇、王玉珏、张众。

# 引　言

本部分定义了八种采用 $n$ 比特分组密码的消息鉴别码算法（MAC 算法）：CBC-MAC、EMAC、ANSI retail MAC、MacDES、CMAC、LMAC、TrCBC、CBCR。

本部分定义的第一个 MAC 算法通常被称作 CBC-MAC。其余七个 MAC 算法是 CBC-MAC 的变种。MAC 算法 2、3、5、6 和 8 在操作的末尾应用了特殊的变换。MAC 算法 4 在操作的起始和末尾各应用了一个特殊的变换。MAC 算法 7 在截取 MAC 值时使用特殊的规则。当 MAC 算法的密钥长度是分组密码密钥长度的两倍的时候，建议使用 MAC 算法 4。MAC 算法 5 和 7 使用加密的次数最少。MAC 算法 5 只需要一次分组密码密钥设置，但需要一个较长的中间密钥。MAC 算法 6 是 MAC 算法 2 的可选变种。MAC 算法 7 和 8 不需要中间密钥和密钥设置，当存储空间受限时，建议使用 MAC 算法 7 和 8。

本部分凡涉及密码算法的相关内容，按国家有关法规实施；凡涉及到采用密码技术解决保密性、完整性、真实性、抗抵赖性需求的须遵循密码相关国家标准和行业标准。

# 信息技术 安全技术 消息鉴别码
# 第1部分：采用分组密码的机制

## 1 范围

GB/T 15852的本部分定义了八种采用分组密码的消息鉴别码（MAC）算法，规定了这八种MAC算法的用户使用要求和一般模型，提供了测试向量和安全性分析。

本部分适用于安全体系结构、过程及应用的安全服务。

## 2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 9387.2-1995 信息处理系统 开放系统互联 基本参考模型 第2部分：安全体系结构

GB/T 15843.1-2017 信息技术 安全技术 实体鉴别 第1部分：总则

GB/T 17964-2008 信息安全技术 分组密码算法的工作模式

GB/T 32907-2016 信息安全技术 SM4分组密码算法

## 3 术语和定义

下列术语和定义适用于本文件。

3.1

**分组 block**

长度为$n$的比特串。

3.2

**密钥 key**

控制密码变换操作的符号序列。

注：密码变换操作，如加密、解密、密码校验函数计算、签名生成、签名验证。

3.3

**明文 plaintext**

未加密的信息。

3.4

**密文 ciphertext**

为隐藏信息内容进行变换后的数据。

3.5

**分组密码密钥 block cipher key**

控制分组密码运算的密钥。

3.6

**$n$比特分组密码 $n$-bit block cipher**

分组长度为$n$比特的分组密码。

3.7

**加密 encryption**

为隐藏数据信息，通过密码算法对数据进行的一种可逆变换过程，并产生密文。

3.8

**解密 decryption**

一个相应的加密过程的逆过程。

3.9

**数据完整性 data integrity**

数据未被非授权地修改或破坏的性质。

3.10

**消息鉴别码（MAC） Message Authentication Code**

利用对称密码技术，以密钥为参数，由消息导出的数据项。任何持有这一密钥的实体，都可利用消息鉴别码检查消息的完整性和始发者。

注：一个MAC有时也称作一个密码校验值。

3.11

**消息鉴别码算法 Message Authentication Code algorithm**

消息鉴别码算法简称MAC算法，其输入为密钥和消息，输出为一个固定长度的比特串，满足下面两个性质：

——对于任何密钥和消息，MAC 算法都能够快速有效地计算。

——对于任何固定的密钥，攻击者在没有获得密钥信息的情况下，即使获得了一些（消息，MAC）对，对任何新的消息预测其 MAC 在计算上是不可行的。

注 1：一个 MAC 算法有时也称作一个密码校验函数。

注 2：计算不可行性依赖于使用者具体的安全要求及其环境。

3.12

**消息鉴别码（MAC）算法密钥 MAC algorithm key**

用于控制消息鉴别码算法运算的密钥。

3.13

**初始变换 initial transformation**

消息鉴别码算法起始时所应用的函数。

3.14

**输出变换 output transformation**
应用在算法中，对迭代操作的输出所进行的变换。

## 4 符号、代号和缩略语

### 4.1 符号和代号

本部分使用下列符号约定：

| | |
|---|---|
| $CT_i$ | 整数 $i$ 的 $n$ 比特的二进制表示。 |
| $D$ | 输入 MAC 算法的数据比特串。 |
| $D_j$ | 填充和分割操作后，分割自数据比特串 $D$ 的分组。 |
| $d_K(C)$ | 使用分组密码 $e$ 和密钥 $K$ 对密文 $C$ 进行解密。 |
| $e_K(P)$ | 使用分组密码 $e$ 和密钥 $K$ 对明文 $P$ 进行加密。 |
| $F$ | 最终迭代。 |
| $G$ | 输出变换的结果。 |
| $g$ | 输出变换，将分组 $H_q$ 映射到分组 $G$ 。 |
| GF($2^n$) | 元素个数为 $2^n$ 的有限域。 |
| $H_0, H_1, ..., H_q$ | MAC 算法运算中的中间变量。 |
| $I$ | 初始变换。 |
| $K, K', K''$ | 分组密码的秘密密钥，长度为 $k$ 比特。 |
| $K_1, K_2$ | 秘密掩码密钥，长度为 $n$ 比特。 |
| $k$ | 分组密码密钥的比特长度。 |
| $k^*$ | MAC 算法密钥的比特长度。 |
| $L$ | 填充方法 3 中表示长度的分组，等价于输入消息长度的二进制表示经左侧填充得到 $n$ 比特分组。 |
| $L_D$ | 数据比特串 $D$ 的比特长度。 |
| $m$ | MAC 值的比特长度。 |
| $n$ | 分组密码的分组长度。 |
| $p_n(x)$ | GF(2) 上的 $n$ 次不可约多项式，即：没有非平凡因子的多项式。 |
| $\tilde{p}_n$ | 长度为 $n$ 的比特串，包含不可约多项式 $p_n(x)$ 最右侧的 $n$ 个系数（对应于 $x^{n-1}, x^{n-2}, ..., x, x^0 = 1$ ）。<br>对于 $n = 128$ ， $p_n(x) = x^{128} + x^7 + x^2 + x + 1$ ， $\tilde{p}_{128} = 0^{120}10000111$ 。 |

对于 $n=64$， $p_n(x)=x^{64}+x^4+x^3+x+1$， $\tilde{p}_{64}=0^{59}11011$。

$q$ 经过填充和分割操作之后，数据比特串 $D$ 的分组个数。

$S$ 长度为 $n$ 的秘密比特串。

$S_1, S_2$ 长度为 $t \cdot n$ 的秘密比特串。

$t$ 不小于 $k/n$ 的最小整数。

$LSB_j(X)$ 比特串 $X$ 最右侧 $j$ 比特串。

$MSB_j(X)$ 比特串 $X$ 最左侧 $j$ 比特串。

multx($T$) 长度为 $n$ 的比特串 $T$ 上的操作，记作 $T*x$，其中 $T$ 是有限域 GF($2^n$) 上的元素，将 $T$ 与 GF($2^n$) 的单项式 $x$ 相乘。计算如下，其中 $T_{n-1}$ 表示 $T$ 的最左侧的比特：

$$\text{multx}(T)=\begin{cases} T=\ 1 & \text{当} T_{n-1}=0\text{时} \\ (T=\ 1)\oplus \tilde{p}_n & \text{当} T_{n-1}=1\text{时} \end{cases}。$$

$X \oplus Y$ 比特串 $X$ 和 $Y$ 的异或值。

$X \| Y$ 按顺序将比特串 $X$ 和 $Y$ 连接所构成的比特串。

$0^n$ $n$ 个零比特组成的比特串。

:= MAC 算法定义中使用的赋值符号。

* 有限域乘法。在多项式表达式中，GF($2^n$) 的每个元素可由次数小于 $n$ 的二进制多项式表示。更明确地，比特串 $A=a_{n-1}...a_2a_1a_0$ 映射到二进制多项式 $a(x)=a_{n-1}x^{n-1}+...+a_2x^2+a_1x+a_0$。有限域 GF($2^n$) 中的乘法，记作 $A*B$，等价于两个多项式的乘积 $a(x)b(x)$ 模一个 $n$ 次不可约多项式 $p_n(x)$。即：$A*B$ 是经 $a(x)$ 和 $b(x)$ 相乘后，除以 $p_n(x)$ 所得的次数不大于 $n-1$ 的余式。其中 $p_n(x)$ 选具有最少非零系数的 $n$ 次不可约多项式中按字典序排列的第一个多项式。对于 $n=128$，$p_n(x)=x^{128}+x^7+x^2+x+1$。

$X=\ 1$ 比特串 $X$ 左移 1 位得到的比特串；如果 $X$ 的长度是 $n$ 比特，则 $X=\ 1$ 表示 $X \| 0$ 的最右侧 $n$ 比特。

$X\ \ 1$ 比特串 $X$ 循环左移 1 位得到的比特串。

$X\ \ 1$ 比特串 $X$ 循环右移 1 位得到的比特串。

## 4.2 缩略语

下列缩略语适用于本文件。

| | | |
|---|---|---|
| CBC | Cipher Block Chaining | 分组密码链接 |
| CTR | Counter Operation Mode | 计数器模式 |
| MAC | Message Authentication Code | 消息鉴别码 |

## 5 用户要求

使用本部分中给出的MAC算法的用户需要选择：

——符合国家管理要求的分组密码算法 $e$；

——从 6.3 中选取一种填充方法；

——从 7 中选取一个 MAC 算法；

——MAC 的比特长度 $m$；

——一个通用的密钥诱导方法；MAC 算法 5 需要此方法，MAC 算法 2、4 和 6 也可能需要。

MAC 的比特长度 $m$ 应是一个正整数并且不大于分组长度 $n$。

如果使用填充方法 3，那么数据串 $D$ 的比特长度应小于 $2^n$。

如果使用 MAC 算法 4，那么数据串填充中的分组数应不小于 2，即：$q \geq 2$。

如果使用 MAC 算法 7，那么 MAC 的比特长度 $m$ 应小于 $n/2$。

对于具体分组密码 $e$、填充方法、MAC 算法、$m$ 的值以及密钥诱导方法（如果需要）的选择超出了本部分所规定的范围。

**注**：上述选择影响MAC算法的安全强度。具体参见附录B。

生成MAC和验证MAC应使用相同的密钥。当数据比特串也被加密，则MAC算法使用的密钥应不同于用作加密的密钥。

**注**：关于密钥管理的信息参见GB/T 17901.1-1999。

本部分中给出的MAC算法的安全性严重依赖于管理密钥所遵循的程序和方法。

MAC算法计算过程中泄露中间值可能造成伪造和（或）密钥恢复攻击（参见附录B）。

## 6 MAC 算法的模型

### 6.1 一般模型

MAC算法的应用需要如下八步操作：

——第 1 步：密钥诱导（可选）；

——第 2 步：消息填充；

——第 3 步：数据分割；

——第 4 步：初始变换 $I$；

——第 5 步：迭代应用分组密码；

——第 6 步：最终迭代 $F$；

——第 7 步：输出变换 $g$；

——第 8 步：截断操作。

其中，第4步至第8步如图1所示。

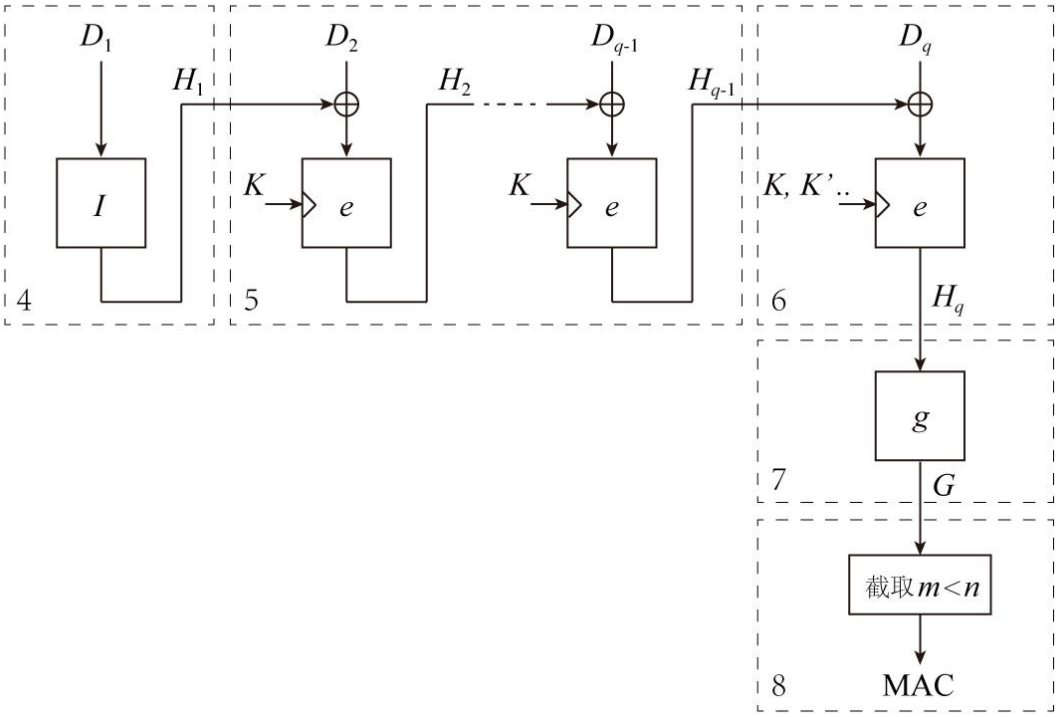


图 1　MAC 算法的第 4、5、6、7 和 8 步操作

## 6.2　密钥诱导（第 1 步）

### 6.2.1　概述

MAC算法5使用密钥诱导算法，由一个分组密码密钥诱导得到两个掩码密钥。MAC算法2、4和6可能需要使用密钥诱导算法，由一个分组密码密钥诱导得到两个分组密码密钥。

本部分规定了两种密钥诱导算法。密钥诱导算法1可被用在本部分所规定的MAC算法2、4和6中。密钥诱导算法2被用在本部分所规定的MAC算法5中。

### 6.2.2　密钥诱导方法 1

密钥诱导方法1由一个分组密码密钥 $K$ 计算得到两个 $k$ 比特的分组密码密钥 $K'$ 和 $K''$。

密钥诱导方法使用GB/T 17964中定义的计数器模式（CTR）。操作如下：

——定义整数 $t$ 为不小于 $k/n$ 的最小正整数。

——定义计数器 $CT_i$（$1 \le i \le 2t$）为在整数 $i$ 的二进制表示左侧填充“0”得到的 $n$ 比特分组，尽可能少填充（甚至不填充）。

——计算长度为 $tn$ 的比特串 $S_1$，使其等于 $e_K(CT_1) \| e_K(CT_2) \| ... \| e_K(CT_t)$，并定义 $K' := MSB_k(S_1)$。

——计算长度为 $tn$ 的比特串 $S_2$，使其等于 $e_K(CT_{t+1}) \| e_K(CT_{t+2}) \| ... \| e_K(CT_{2t})$，并定义 $K'' := MSB_k(S_2)$。

### 6.2.3　密钥诱导方法 2

密钥诱导方法2由一个分组密码密钥 $K$ 计算得到两个 $n$ 比特的掩码密钥 $K_1$ 和 $K_2$。操作如下：

——首先，计算 $n$ 比特的秘密比特串 $S$：$S := e_K(0^n)$。

——其次，由 $S$ 计算得到掩码密钥 $K_1$：$K_1 := \mathrm{multx}(S)$。

——最后，由 $K_1$ 计算得到掩码密钥 $K_2$： $K_2 := \mathrm{multx}(K_1)$。

### 6.3 消息填充（第 2 步）

#### 6.3.1 概述

在这一步骤中，用额外的比特串作为前缀或后缀对数据比特串 $D$ 进行填充，使得填充后的数据比特串的长度是 $n$ 的整数倍。根据选择的填充方法，填充比特串只用来计算MAC，所以这些填充比特串不必随原消息存储或发送。MAC的验证者应知道填充比特串是否已经被存储或发送，以及使用的是何种填充方法。

本部分规定了四种填充方法。填充方法1、2和3可被用在本部分所规定的MAC算法1、2、3、4和6中。填充方法4被用在本部分所规定的MAC算法5、7和8中。

#### 6.3.2 填充方法 1

在数据比特串 $D$ 的右侧填充“0”，尽可能少填充（甚至不填充），使填充后比特串的长度是 $n$ 的正整数倍。

注 1：面对简单伪造攻击，使用填充方法 1 的 MAC 算法可能是不安全的。具体参见附录 B。

注 2：如果数据比特串是空串，那么填充方法 1 规定对其填充 $n$ 个“0”。

#### 6.3.3 填充方法 2

在数据比特串 $D$ 的右侧填充一个比特“1”，然后在所得到的比特串右侧填充“0”，尽可能少填充（甚至不填充），使填充后的比特串的长度是 $n$ 的正整数倍。

注：如果数据比特串是空串，那么填充方法2规定对其填充一个“1”，然后在其右侧填充 $n-1$ 个“0”。

#### 6.3.4 填充方法 3

在数据比特串 $D$ 的右侧填充“0”，尽可能少填充（甚至不填充），使填充后比特串的长度是 $n$ 的正整数倍。然后在所得到的数据比特串左侧填充一个分组 $L$。分组 $L$ 由尽可能少的“0”和数据比特串 $D$ 的长度 $L_D$ 的二进制表示组成，其中位于 $L_D$ 二进制表示的左侧的“0”尽可能少，且使 $L$ 的长度为 $n$ 比特。$L$ 最右端的比特和 $L_D$ 的二进制表示中的最低位相对应。

注 1：如果在计算 MAC 之前不知数据比特串的长度，则填充方法 3 不适用。

注 2：如果数据比特串是空串，那么填充方法 3 规定对其填充 $n$ 个“0”，然后在其左侧填充一个由 $n$ 个“0”组成的分组 $L$。

#### 6.3.5 填充方法 4

如果输入MAC算法的数据比特串 $D$ 的比特长度是 $n$ 的正整数倍，则不需要填充。否则，在数据比特串 $D$ 的右侧填充一个“1”比特，然后在所得到的比特串右侧填充“0”，尽可能少填充（甚至不填充），使填充后的比特串的长度是 $n$ 的正整数倍。

注：如果数据比特串是空串，那么填充方法4规定对其填充一个“1”，然后在其左侧填充 $n-1$ 个“0”。

### 6.4 数据分割（第 3 步）

把填充后的数据比特串分割成 $q$ 个 $n$ 比特的分组 $D_1, D_2, ..., D_q$。这里 $D_1$ 表示填充后比特串的第一个 $n$ 比特，$D_2$ 表示随后的 $n$ 个比特，以此类推。

### 6.5 初始变换（第 4 步）

### 6.5.1 概述

初始变换 $I$ 用来处理填充后比特串的第一个 $n$ 比特分组 $D_1$ 以得到 $H_1$。

本部分规定了三种初始变换。初始变换1被用在本部分所规定的MAC算法1、2、3、5、6和7中。初始变换2被用在本部分所规定的MAC算法4中。初始变换3被用在本部分所规定的MAC算法8中。

### 6.5.2 初始变换 1

初始变换1需要一个分组密码密钥 $K$。按照如下的方法使用密钥 $K$ 和分组密码 $e$ 计算 $H_1$：

$$H_1 \coloneqq e_K(D_1)。$$

### 6.5.3 初始变换 2

初始变换2需要两个分组密码密钥 $K$ 和 $K''$。按照如下的方法使用密钥 $K$ 和 $K''$，以及分组密码 $e$ 计算 $H_1$：

$$H_1 \coloneqq e_{K''}(e_K(D_1))。$$

### 6.5.4 初始变换 3

初始变换3需要一个分组密码密钥 $K$。按照如下的方法使用密钥 $K$，以及分组密码 $e$ 计算 $H_1$：

$$H_1 \coloneqq e_K(D_1 \oplus H_0)，$$

其中 $H_0 \coloneqq e_K(0^n)$。

## 6.6 迭代应用分组密码（第 5 步）

对比特串 $D_i$ 和 $H_{i-1}$ 的异或值迭代应用密钥为 $K$ 的分组密码 $e$，计算得到分组 $H_2, H_3, ..., H_{q-1}$：

$$H_i \coloneqq e_K(D_i \oplus H_{i-1}),\ i = 2, ..., q-1。$$

如果 $q = 2$，那么第5步省略。

## 6.7 最终迭代（第 6 步）

### 6.7.1 概述

最终迭代 $F$ 用来处理填充后比特串的最后一个分组 $D_q$ 以得到分组 $H_q$。

本部分规定了四种最终迭代。最终迭代1被用在本部分所规定的MAC算法1、2、3、4和7中。最终迭代2被用在本部分所规定的MAC算法6中。最终迭代3被用在本部分所规定的MAC算法5中。最终迭代4被用在本部分所规定的MAC算法8中。

### 6.7.2 最终迭代 1

最终迭代1使用与第5步相同的分组密码密钥 $K$，以及分组密码 $e$ 计算 $H_q$：

$$H_q \coloneqq e_K(D_q \oplus H_{q-1})。$$

### 6.7.3 最终迭代 2

最终迭代2使用分组密码密钥 $K'$（不同于第5步使用的分组密码密钥 $K$），以及分组密码 $e$ 计算 $H_q$：

$$H_q \coloneqq e_{K'}(D_q \oplus H_{q-1})。$$

### 6.7.4 最终迭代 3

最终迭代3使用与第5步相同的分组密码密钥 $K$ 和两个 $n$ 比特的掩码密钥 $K_1$ 和 $K_2$。根据填充操作将输入与密钥 $K_1$ 或 $K_2$ 异或，再应用密钥为 $K$ 的分组密码 $e$，计算得到 $H_q$。

依照填充方法4，如果输入MAC算法的消息的比特长度是 $n$ 的正整数倍，则：

$$H_q := e_K(D_q \oplus H_{q-1} \oplus K_1)，$$

否则

$$H_q := e_K(D_q \oplus H_{q-1} \oplus K_2)。$$

### 6.7.5 最终迭代 4

最终迭代4使用与第5步相同的分组密码密钥 $K$。根据填充操作将输入循环右移或左移1位。

依照填充方法4，如果输入MAC算法的消息的比特长度是 $n$ 的正整数倍，则：

$$H_q := e_K((D_q \oplus H_{q-1}) \quad 1)，$$

否则

$$H_q := e_K((D_q \oplus H_{q-1}) \quad 1)。$$

## 6.8 输出变换（第 7 步）

### 6.8.1 概述

输出变换 $g$ 用来处理第6步得到的结果 $H_q$。

本部分规定了三种输出变换。输出变换1被用在本部分所规定的MAC算法1、5、6、7和8中。输出变换2被用在本部分所规定的MAC算法2和4中。输出变换3被用在本部分所规定的MAC算法3中。

### 6.8.2 输出变换 1

输出变换1是恒等变换：

$$G := H_q。$$

### 6.8.3 输出变换 2

输出变换2对 $H_q$ 应用密钥为 $K'$ 的分组密码 $e$，即：

$$G := e_{K'}(H_q)。$$

### 6.8.4 输出变换 3

输出变换3对 $H_q$ 应用密钥为 $K'$ 的分组密码（解密操作）$d$，对于得到的结果再应用密钥为 $K$ 的分组密码 $e$，即：

$$G := e_K(d_{K'}(H_q))。$$

### 6.9 截断操作（第 8 步）

#### 6.9.1 概述

截断操作用来处理第7步得到的结果 $G$ 以得到MAC值。

本部分规定了两种截断操作。截断操作1被用在本部分所规定的MAC算法1、2、3、4、5、6和8中。截断操作2被用在本部分所规定的MAC算法7中。

#### 6.9.2 截断操作 1

截断操作1截取 $G$ 最左侧的 $m$ 比特作为MAC值，即：

$$\mathrm{MAC} := MSB_m(G) \text{ 。}$$

#### 6.9.3 截断操作 2

截断操作2根据填充操作截取 $G$ 最左侧或最右侧的 $m$ 比特作为MAC值。

依照填充方法4，如果输入MAC算法的消息的比特长度是 $n$ 的正整数倍，则：

$$\mathrm{MAC} := MSB_m(G) \text{ ，}$$

否则

$$\mathrm{MAC} := LSB_m(G) \text{ 。}$$

## 7 MAC 算法

### 7.1 概述

本部分规定了八种MAC算法。每种MAC算法明确规定了初始变换、最终迭代、输出变换和截断操作。

### 7.2 MAC 算法 1（CBC-MAC）

MAC算法1使用初始变换1、最终迭代1、输出变换1和截断操作1。MAC算法密钥就是分组密码密钥 $K$ 。

MAC算法1如图2所示。

MAC算法1可使用6.3中的填充方法1、2或3。

注 1：填充方法的选择影响 MAC 算法的安全性。具体参见资料性附录 B。

注 2：MAC 算法 1 可能遭受异或伪造攻击（参见附录 B）。因此 MAC 算法 1 仅适用于异或伪造攻击不可行的环境，例如，消息长度固定。

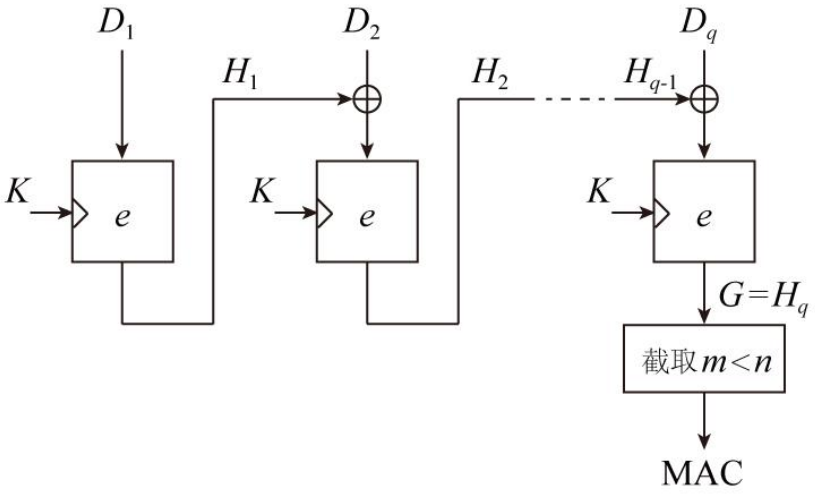


图 2　MAC 算法 1（CBC-MAC）

## 7.3　MAC 算法 2（EMAC）

MAC算法2使用初始变换1、最终迭代1、输出变换2和截断操作1。MAC算法密钥由两个分组密码密钥 $K$ 和 $K'$ 组成。$K$ 和 $K'$ 的值可从一个共同的主密钥（一个分组密码密钥）通过密钥诱导方法生成，应满足 $K$ 和 $K'$ 以高概率不相同。

注 1：MAC 算法 2 通常被称作 EMAC[25]。

注 2：密钥诱导方法 1 是由一个共同的主密钥诱导出 $K$ 和 $K'$ 的一个例子。

注 3：若 $K=K'$，一个简单的异或伪造攻击就能够攻击 EMAC，具体参见资料性附录 B。

注 4：若 $K$ 和 $K'$ 相互独立，MAC 算法 2 针对密钥恢复攻击的安全强度低于 MAC 算法 2 采用的密钥长度所应提供的安全强度，具体参见资料性附录 B。

MAC算法2如图3所示。

MAC算法2可使用6.3中的填充方法1、2或3。

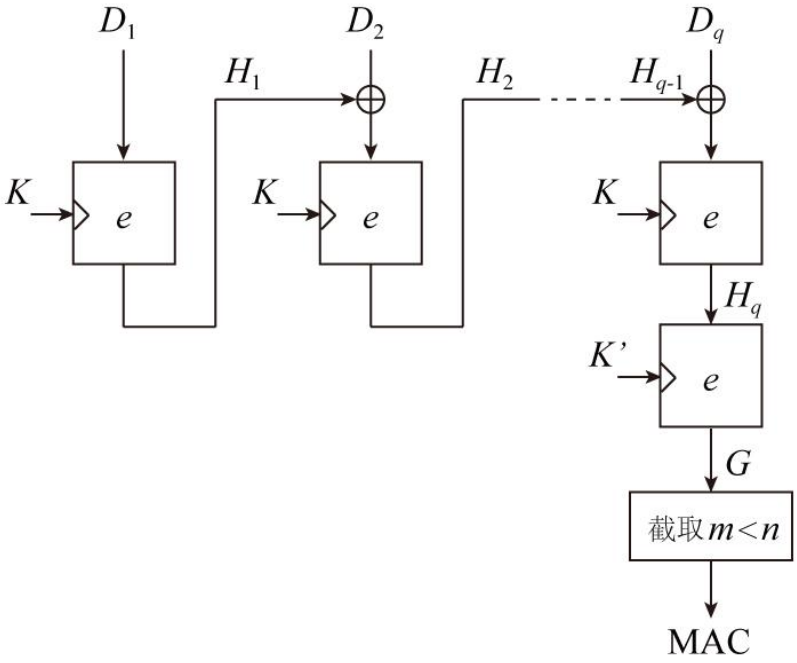


**图 3　MAC 算法 2（EMAC）**

**注 5：**填充方法的选择影响 MAC 算法的安全性。具体参见资料性附录 B。

**注 6：**如果 MAC 算法 2 与计算（公开）密钥标识 $S=e_K(0^n)$ 的算法结合使用，例如 X9.24[11] ，则 MAC 算法 2 将遭受异或伪造攻击（参见附录 B）。在这种情况下，算法应仅用于异或伪造攻击不可行的环境，例如，消息长度固定。

### 7.4　MAC 算法 3（ANSI retail MAC）

MAC算法3使用初始变换1、最终迭代1、输出变换3和截断操作1。MAC算法密钥由两个分组密码密钥 $K$ 和 $K'$ 组成。 $K$ 和 $K'$ 应独立选取。若 $K=K'$， MAC算法3和MAC算法1一致。

MAC算法3如图4所示。

MAC算法3可使用6.3中的填充方法1、2或3。

**注 1：**MAC 算法 3 通常被称作 ANSI retail MAC[10]。

**注 2：**填充方法的选择影响 MAC 算法的安全性。具体参见资料性附录 B。

**注 3：**如果 MAC 算法 3 与计算（公开）密钥标识 $S=e_K(0^n)$ 的算法结合使用，例如 X9.24[11],则 MAC 算法 3 将遭受异或伪造攻击（参见附录 B）。在这种情况下，算法应仅用于异或伪造攻击不可行的环境，例如，消息长度固定。

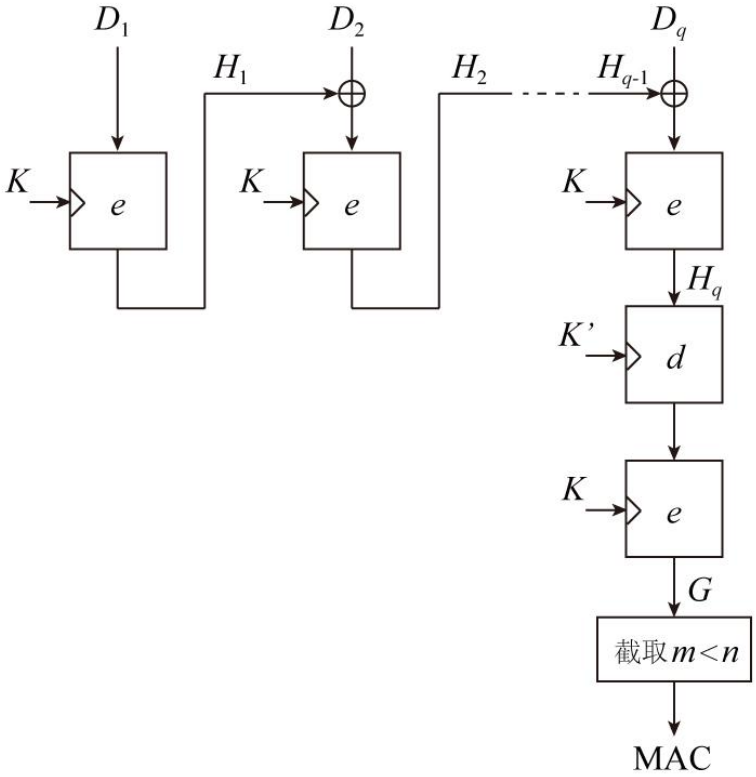


图 4　MAC 算法 3（ANSI retail MAC）

## 7.5　MAC 算法 4（MacDES）

MAC算法4使用初始变换2、最终迭代1、输出变换2和截断操作1。MAC算法密钥由两个分组密码密钥 $K$ 和 $K'$ 组成，$K$ 和 $K'$ 应独立选取。另外，第三个分组密码密钥 $K''$ 由 $K'$ 通过密钥诱导方法得出。密钥 $K$、$K'$ 和 $K''$ 应互不相同。分组密码密钥 $K$ 和 $K''$ 用于初始变换2，分组密码密钥 $K'$ 用于输出变换2。

注 1：MAC 算法 4 在[22]中被提出，因提出时采用 DES（ISO/IEC 18033-3:2005 的附录 A 和 ANSI X3.92[8]中规范的分组密码）而被称作 MacDES。

注 2：对 $K'$ 从第一个 4 比特组开始，每隔 4 比特交替取补和不变即是由 $K'$ 诱导出 $K''$ 的一个例子。这个例子中，$K'$ 和 $K''$ 不是相互独立且随机的。另外一个例子是用密钥诱导方法 1 由一个公共的主密钥生成 $K'$ 和 $K''$。

MAC算法4如图5所示。

MAC算法4可使用6.3中的填充方法1、2或3。

注 3：填充方法的选择影响 MAC 算法的安全性。具体参见资料性附录 B。

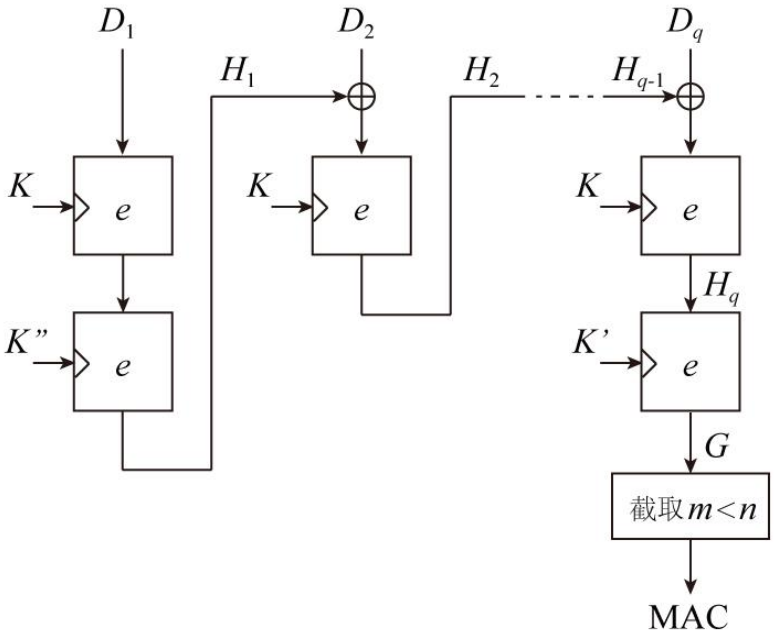


图 5　MAC 算法 4（MacDES）

## 7.6　MAC 算法 5（CMAC）

MAC算法5使用密钥诱导方法2、初始变换1、最终迭代3、输出变换1和截断操作1。MAC算法密钥由一个分组密码密钥 $K$ 组成。MAC算法5使用填充方法4。最终迭代3中使用的掩码密钥 $K_1$ 和 $K_2$ 是使用密钥诱导方法2由MAC算法密钥 $K$ 得到的。

MAC算法5如图6所示，其中 $K_i = K_1$ 或 $K_2$ 。

注 1：MAC 算法 5 在[20]中提出，通常被称作 OMAC1[20]或 CMAC[12]。

注 2：如果 MAC 算法 5 与计算（公开）密钥标识 $S = e_K(0^n)$ 的算法结合使用，例如 X9.24[11],则 MAC 算法 5 将遭受异或伪造攻击（参见附录 B）。在这种情况下，算法应仅用于异或伪造攻击不可行的环境，例如，消息长度固定。

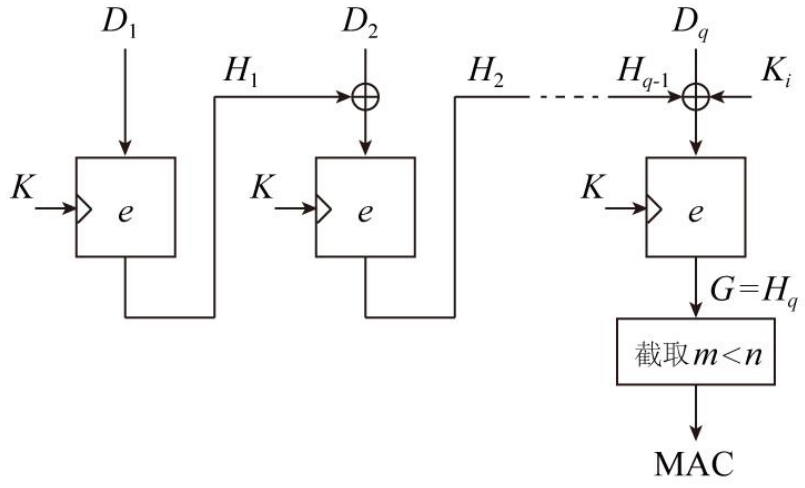


图 6　MAC 算法 5（CMAC）

## 7.7 MAC 算法 6（LMAC）

MAC算法6使用初始变换1、最终迭代2、输出变换1和截断操作1。MAC算法密钥包含两个分组密码密钥 $K$ 和 $K'$。$K$ 和 $K'$ 的值可从一个共同的主密钥（一个分组密码密钥）通过密钥诱导方法生成，应满足 $K$ 和 $K'$ 以高概率不相同。

注 1：MAC 算法 6 通常被称作 LMAC。

注 2：密钥诱导方法 1 是由一个共同的主密钥诱导出 $K$ 和 $K'$ 的一个例子。

注 3：若 $K = K'$，一个简单的异或伪造攻击就能够攻击 MAC 算法 6，具体参见资料性附录 B。

注 4：若 $K$ 和 $K'$ 相互独立，MAC 算法 6 针对密钥恢复攻击的安全强度低于 MAC 算法 6 采用的密钥长度所应提供的安全强度，具体参见资料性附录 B。

MAC算法6如图7所示。

MAC算法6可使用6.3中的填充方法1、2和3。

注 5：填充方法的选择影响 MAC 算法的安全性。具体参见资料性附录 B。

注 6：如果 MAC 算法 6 与计算（公开）密钥标识 $S = e_K(0^n)$ 的算法结合使用，例如 X9.24[11]，则 MAC 算法 6 将遭受异或伪造攻击（参见附录 B）。在这种情况下，算法应仅用于异或伪造攻击不可行的环境，例如，消息长度固定。

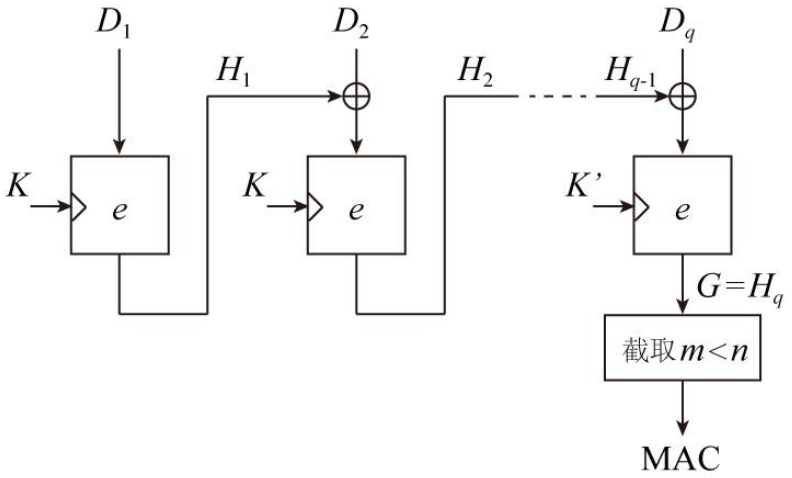


图 7 MAC 算法 6（LMAC）

## 7.8 MAC 算法 7（TrCBC）

MAC算法7使用初始变换1、最终迭代1、输出变换1和截断操作2。MAC算法密钥就是分组密码密钥 $K$。MAC算法7使用填充方法4。

注：MAC算法7通常被称作TrCBC[29]。

MAC算法7如图8所示。

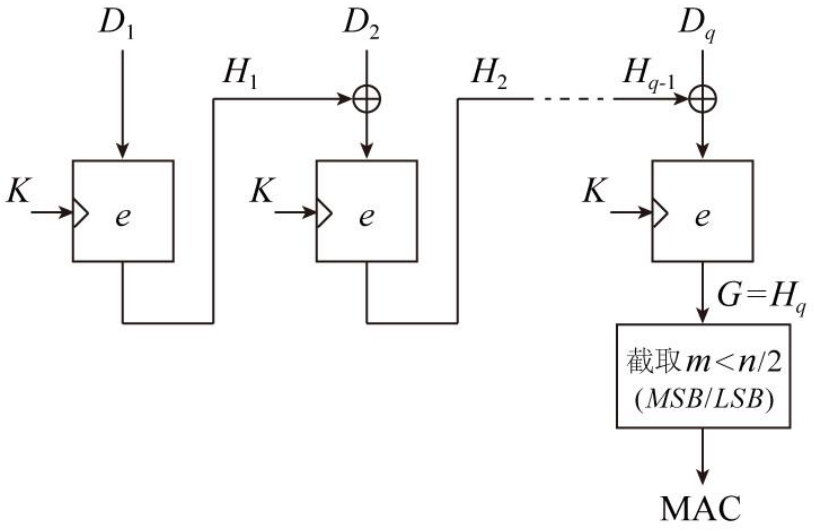


图 8　MAC 算法 7（TrCBC）

## 7.9　MAC 算法 8（CBCR）

MAC算法8使用初始变换3、最终迭代4、输出变换1和截断操作1。MAC算法密钥就是分组密码密钥 $K$ 。MAC算法8使用填充方法4。

注：MAC算法8通常被称作CBCR[30]。

MAC算法8如图9所示。

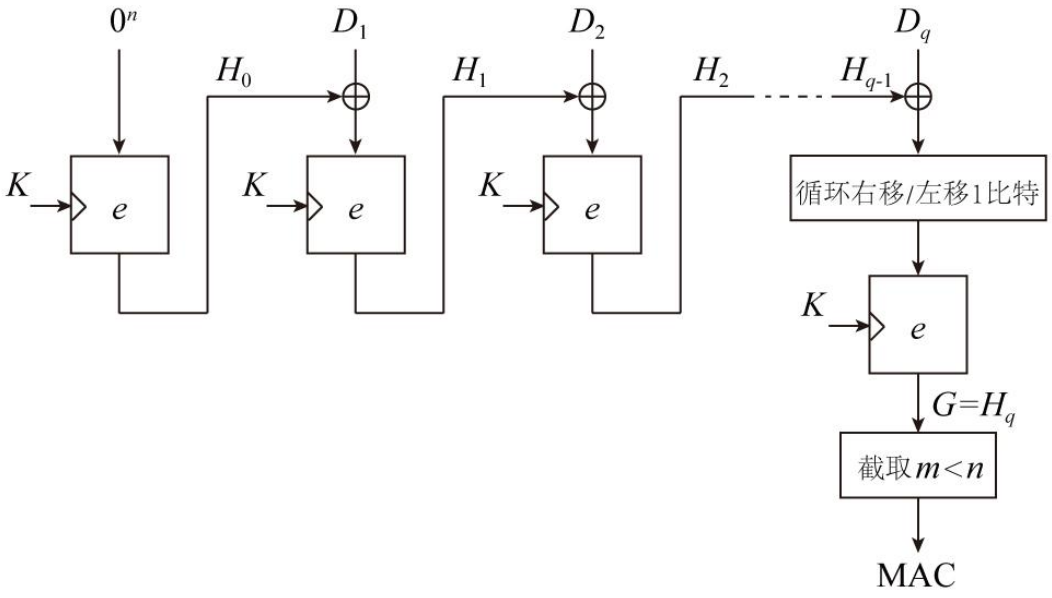


图 9　MAC 算法 8（CBCR）

# 附 录 A
（资料性附录）
测试向量

## A.1 概述

本附录提供了使用GB/T 32907-2016中定义的SM4分组密码算法生成MAC的过程示例。

对于MAC算法，明文是GB/T 1988-1998规定的七位编码字符：数据比特串1：“This_is_the_test_message_for_mac”，数据比特串2为：“This_is_the_test_message_”，其中“_”表示一个空格。所有的MAC值和密钥值都是用16进制表示。

对于数据比特串1，分别经过填充方法1-4后得到的结果如下：

——填充方法 1：$q=2$

| | |
|---|---|
| $D_1$ | 54 68 69 73 20 69 73 20 74 68 65 20 74 65 73 74 |
| $D_2$ | 20 6D 65 73 73 61 67 65 20 66 6F 72 20 6D 61 63 |

——填充方法 2：$q=3$

| | |
|---|---|
| $D_1$ | 54 68 69 73 20 69 73 20 74 68 65 20 74 65 73 74 |
| $D_2$ | 20 6D 65 73 73 61 67 65 20 66 6F 72 20 6D 61 63 |
| $D_3$ | 80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |

——填充方法 3：$q=3$

| | |
|---|---|
| $D_1$ | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 |
| $D_2$ | 54 68 69 73 20 69 73 20 74 68 65 20 74 65 73 74 |
| $D_3$ | 20 6D 65 73 73 61 67 65 20 66 6F 72 20 6D 61 63 |

——填充方法 4：$q=2$

| | |
|---|---|
| $D_1$ | 54 68 69 73 20 69 73 20 74 68 65 20 74 65 73 74 |
| $D_2$ | 20 6D 65 73 73 61 67 65 20 66 6F 72 20 6D 61 63 |

对于数据比特串2，分别经过填充方法1-4后得到的结果如下：

——填充方法 1：$q=2$

| | |
|---|---|
| $D_1$ | 54 68 69 73 20 69 73 20 74 68 65 20 74 65 73 74 |
| $D_2$ | 20 6D 65 73 73 61 67 65 20 00 00 00 00 00 00 00 |

——填充方法 2：$q=2$

| | |
|---|---|
| $D_1$ | 54 68 69 73 20 69 73 20 74 68 65 20 74 65 73 74 |
| $D_2$ | 20 6D 65 73 73 61 67 65 20 80 00 00 00 00 00 00 |

——填充方法 3：$q=3$

| | |
|---|---|
| $D_1$ | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 C8 |
| $D_2$ | 54 68 69 73 20 69 73 20 74 68 65 20 74 65 73 74 |
| $D_3$ | 20 6D 65 73 73 61 67 65 20 00 00 00 00 00 00 00 |

——填充方法 4：$q=2$

| | |
|---|---|
| $D_1$ | 54 68 69 73 20 69 73 20 74 68 65 20 74 65 73 74 |
| $D_2$ | 20 6D 65 73 73 61 67 65 20 80 00 00 00 00 00 00 |

## A.2 MAC算法 1（CBC-MAC）

这里使用的密钥是 $K$ = 01 23 45 67 89 AB CD EF FE DC BA 98 76 5 4 32 10 （16进制）。MAC的比特长度 $m$ 等于64。

——使用数据比特串 1 和填充方法 1

| | |
|---|---|
| 密钥 $K$ | 01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10 |
| $H_1$ | 45 FF A9 48 60 5F 52 E8 F4 EF 21 D5 5C D8 F8 0C |
| $D_2 \oplus H_1$ | 65 92 CC 3B 13 3E 35 8D D4 89 4E A7 7C B5 99 6F |
| $G = H_2$ | 16 E0 29 04 EF B7 65 B7 06 45 9C 9E DA BD B5 19 |

MAC=16 E0 29 04 EF B7 65 B7

——使用数据比特串 1 和填充方法 2

| | |
|---|---|
| 密钥 $K$ | 01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10 |
| $H_1$ | 45 FF A9 48 60 5F 52 E8 F4 EF 21 D5 5C D8 F8 0C |
| $D_2 \oplus H_1$ | 65 92 CC 3B 13 3E 35 8D D4 89 4E A7 7C B5 99 6F |
| $H_2$ | 16 E0 29 04 EF B7 65 B7 06 45 9C 9E DA BD B5 19 |
| $D_3 \oplus H_2$ | 96 E0 29 04 EF B7 65 B7 06 45 9C 9E DA BD B5 19 |
| $G = H_3$ | 4B 65 53 AF 3C 4E 27 44 84 12 31 5A C7 84 95 35 |

MAC=4B 65 53 AF 3C 4E 27 44

——使用数据比特串 1 和填充方法 3

| | |
|---|---|
| 密钥 $K$ | 01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10 |
| $H_1$ | 03 FE 50 C5 56 A3 DB 0F CA AC F5 A7 C1 C5 0C A9 |
| $D_2 \oplus H_1$ | 57 96 39 B6 76 CA A8 2F BE C4 90 87 B5 A0 7F DD |
| $H_2$ | A9 15 8A 4B 7B C6 F7 DB 00 23 8D 04 DC 6A 94 A4 |
| $D_3 \oplus H_2$ | 89 78 EF 38 08 A7 90 BE 20 45 E2 76 FC 07 F5 C7 |
| $G = H_3$ | 71 AF 7E 45 53 40 4C BC C4 F2 97 3C DB D0 F0 63 |

MAC=71 AF 7E 45 53 40 4C BC

——使用数据比特串 2 和填充方法 1

| | |
|---|---|
| 密钥 $K$ | 01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10 |
| $H_1$ | 45 FF A9 48 60 5F 52 E8 F4 EF 21 D5 5C D8 F8 0C |
| $D_2 \oplus H_1$ | 65 92 CC 3B 13 3E 35 8D D4 EF 21 D5 5C D8 F8 0C |
| $G = H_2$ | BA 89 E4 5F E8 AB F2 42 E2 6C E0 32 AD 00 7C 09 |

MAC=BA 89 E4 5F E8 AB F2 42

——使用数据比特串 2 和填充方法 2

| | |
|---|---|
| 密钥 $K$ | 01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10 |

| | |
|---|---|
| $H_1$ | 45 FF A9 48 60 5F 52 E8 F4 EF 21 D5 5C D8 F8 0C |
| $D_2 \oplus H_1$ | 65 92 CC 3B 13 3E 35 8D D4 6F 21 D5 5C D8 F8 0C |
| $G = H_2$ | 42 1A D1 69 0A A1 52 E2 84 6F A2 A5 D8 34 45 A9 |

MAC=42 1A D1 69 0A A1 52 E2

——使用数据比特串 2 和填充方法 3

| | |
|---|---|
| 密钥 $K$ | 01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10 |
| $H_1$ | 42 39 BB 2B 9A A0 09 0B E0 D3 48 17 C7 2B 1B C8 |
| $D_2 \oplus H_1$ | 16 51 D2 58 BA C9 7A 2B 94 BB 2D 37 B3 4E 68 BC |
| $H_2$ | CD F9 25 C0 AF 83 15 88 93 76 D0 68 D1 C8 25 3A |
| $D_3 \oplus H_2$ | ED 94 40 B3 DC E2 72 ED B3 76 D0 68 D1 C8 25 3A |
| $G = H_3$ | 6A 4A 86 F5 B5 E4 68 DA D2 7D F2 5F B9 D9 BE 16 |

MAC=6A 4A 86 F5 B5 E4 68 DA

### A.3 MAC算法 2（EMAC）

这里使用两个密钥，$K = 01\ 23\ 45\ 67\ 89\ AB\ CD\ EF\ FE\ DC\ BA\ 98\ 76\ 5\ 4\ 32\ 10$（16进制），$K' = 41\ 49\ D2\ AD\ ED\ 94\ 56\ 68\ 1E\ C8\ B5\ 11\ D9\ E\ 7\ EE\ 04$。诱导密钥 $K'$ 由 $K$ 从第一个4比特组开始，每隔4比特交替取补和不变得到。MAC的比特长度 $m$ 等于64。

前 $q$ 步操作和CBC-MAC一致，唯一的不同是EMAC使用输出变换2。

——使用数据比特串 1 和填充方法 1

| | |
|---|---|
| 密钥 $K'$ | 41 49 D2 AD ED 94 56 68 1E C8 B5 11 D9 E7 EE 04 |
| $G$ | 1E 9A 71 D3 BC 92 DF A7 E5 00 D2 0A 0B 09 41 10 |

MAC=1E 9A 71 D3 BC 92 DF A7

——使用数据比特串 1 和填充方法 2

| | |
|---|---|
| 密钥 $K'$ | 41 49 D2 AD ED 94 56 68 1E C8 B5 11 D9 E7 EE 04 |
| $G$ | E4 23 E3 55 99 AF D9 48 AE C5 0B DE E8 38 E9 EA |

MAC=E4 23 E3 55 99 AF D9 48

——使用数据比特串 1 和填充方法 3

| | |
|---|---|
| 密钥 $K'$ | 41 49 D2 AD ED 94 56 68 1E C8 B5 11 D9 E7 EE 04 |
| $G$ | 40 03 BA 1B 6A DC 53 A8 26 E8 2F CE A1 6A FA AC |

MAC=40 03 BA 1B 6A DC 53 A8

——使用数据比特串 2 和填充方法 1

| | |
|---|---|
| 密钥 $K'$ | 41 49 D2 AD ED 94 56 68 1E C8 B5 11 D9 E7 EE 04 |
| $G$ | 4E C3 C7 FA CF AA C6 07 C3 DD E5 CE B5 03 1C C8 |

MAC=4E C3 C7 FA CF AA C6 07

——使用数据比特串 2 和填充方法 2

| | |
|---|---|
| 密钥 $K'$ | 41 49 D2 AD ED 94 56 68 1E C8 B5 11 D9 E7 EE 04 |
| $G$ | F0 26 25 CE AD 00 8D 4E FB F3 F0 B2 B0 C2 A7 5B |

MAC=F0 26 25 CE AD 00 8D 4E

——使用数据比特串 2 和填充方法 3

| | |
|---|---|
| 密钥$K'$ | 41 49 D2 AD ED 94 56 68 1E C8 B5 11 D9 E7 EE 04 |
| $G$ | FF D5 F1 F2 E5 ED A5 CB F4 02 D6 5A 5B 0B 19 53 |

MAC=FF D5 F1 F2 E5 ED A5 CB

## A.4 MAC算法 3（ANSI retail MAC）

这里使用的两个密钥是 $K = 01\ 23\ 45\ 67\ 89\ AB\ CD\ EF\ FE\ DC\ BA\ 98\ 76\ 54\ 32\ 10$ （16进制）和 $K' = 41\ 49\ D2\ AD\ ED\ 94\ 56\ 68\ 1E\ C8\ B5\ 11\ D9\ E7\ EE\ 04$ （16进制）。MAC的比特长度 $m$ 等于64。

前 $q$ 步操作和MAC算法1一致，唯一的不同是MAC算法3使用输出变换3。

——使用数据比特串 1 和填充方法 1

| | |
|---|---|
| 密钥$K'$ | 41 49 D2 AD ED 94 56 68 1E C8 B5 11 D9 E7 EE 04 |
| $d$ 的输出 | 0C C6 7A FF AF C8 80 C9 05 B9 A9 01 05 DF DA 8D |
| $G$ | 27 63 21 1B 2B CA F7 19 34 90 E4 BD 59 62 AA 67 |

MAC=27 63 21 1B 2B CA F7 19

——使用数据比特串 1 和填充方法 2

| | |
|---|---|
| 密钥$K'$ | 41 49 D2 AD ED 94 56 68 1E C8 B5 11 D9 E7 EE 04 |
| $d$ 的输出 | 7D 5F 48 1C 87 58 55 7B 9A 17 6E 73 4F 5F C8 57 |
| $G$ | 51 E9 92 8C 22 38 33 0C 32 31 B8 75 2A 9A FD 7F |

MAC=51 E9 92 8C 22 38 33 0C

——使用数据比特串 1 和填充方法 3

| | |
|---|---|
| 密钥$K'$ | 41 49 D2 AD ED 94 56 68 1E C8 B5 11 D9 E7 EE 04 |
| $d$ 的输出 | 5B 76 BD 43 4B A8 85 D0 3A 69 A7 F4 2C 33 CF 52 |
| $G$ | 7C D4 8C 42 42 E4 55 75 E5 1A AF 0D CC 7A 20 8C |

MAC=7C D4 8C 42 42 E4 55 75

——使用数据比特串 2 和填充方法 1

| | |
|---|---|
| 密钥$K'$ | 41 49 D2 AD ED 94 56 68 1E C8 B5 11 D9 E7 EE 04 |
| $d$ 的输出 | 6E 67 77 0B 9F 32 B3 8A 17 F2 40 92 49 DA D5 A6 |
| $G$ | E3 2D 99 A6 89 C0 52 59 60 E1 8D 53 AA 73 0F 33 |

MAC=E3 2D 99 A6 89 C0 52 59

——使用数据比特串 2 和填充方法 2

| | |
|---|---|
| 密钥$K'$ | 41 49 D2 AD ED 94 56 68 1E C8 B5 11 D9 E7 EE 04 |
| $d$ 的输出 | 06 21 B9 BE 7C 51 64 4C 4F 7D 3A F8 B1 18 EF 38 |
| $G$ | 19 72 47 22 9C E9 D7 B6 AE 40 5B F8 85 B2 70 57 |

MAC=19 72 47 22 9C E9 D7 B6

——使用数据比特串 2 和填充方法 3

| | |
|---|---|
| 密钥$K'$ | 41 49 D2 AD ED 94 56 68 1E C8 B5 11 D9 E7 EE 04 |

| | |
|---|---|
| $d$ 的输出 | 6D 13 AD 5D 4C EE C5 08 96 A4 4F 48 B2 C2 B9 09 |
| $G$ | 3C 43 0F 1E A4 3B 54 0C 68 45 7E 24 9C 46 F1 DB |

MAC=3C 43 0F 1E A4 3B 54 0C

## A.5 MAC算法4（MacDES）

这里使用两个密钥，$K$ = 01 23 45 67 89 AB CD EF FE DC BA 98 76 5 4 32 10 （16进制）和$K'$ = 41 49 D2 AD ED 94 56 68 1E C8 B5 11 D9 E 7 EE 04 （16进制）。诱导密钥$K''$由$K'$从第一个4比特组开始，每隔4比特交替取补和不变得到。MAC的比特长度$m$等于64。

——使用数据比特串 1 和填充方法 1

| | |
|---|---|
| 密钥$K$ | 01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10 |
| 密钥$K'$ | 41 49 D2 AD ED 94 56 68 1E C8 B5 11 D9 E7 EE 04 |
| 密钥$K''$ | B1 B9 22 5D 1D 64 A6 98 EE 38 45 E1 29 17 1E F4 |
| $e_K(D_1)$ | 45 FF A9 48 60 5F 52 E8 F4 EF 21 D5 5C D8 F8 0C |
| $H_1$ | BA 52 E9 0F 9A 59 91 F0 C9 85 C4 56 69 9F 8D 27 |
| $D_2 \oplus H_1$ | 9A 3F 8C 7C E9 38 F6 95 E9 E3 AB 24 49 F2 EC 44 |
| $H_2$ | CC FD E0 7B 09 96 37 9A 99 F0 4B 68 DE 13 0E 59 |
| $G$ | DD 10 52 A7 AF E8 99 9B BE 31 90 64 3E CF 99 69 |

MAC=DD 10 52 A7 AF E8 99 9B

——使用数据比特串 1 和填充方法 2

| | |
|---|---|
| 密钥$K$ | 01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10 |
| 密钥$K'$ | 41 49 D2 AD ED 94 56 68 1E C8 B5 11 D9 E7 EE 04 |
| 密钥$K''$ | B1 B9 22 5D 1D 64 A6 98 EE 38 45 E1 29 17 1E F4 |
| $e_K(D_1)$ | 45 FF A9 48 60 5F 52 E8 F4 EF 21 D5 5C D8 F8 0C |
| $H_1$ | BA 52 E9 0F 9A 59 91 F0 C9 85 C4 56 69 9F 8D 27 |
| $D_2 \oplus H_1$ | 9A 3F 8C 7C E9 38 F6 95 E9 E3 AB 24 49 F2 EC 44 |
| $H_2$ | CC FD E0 7B 09 96 37 9A 99 F0 4B 68 DE 13 0E 59 |
| $D_3 \oplus H_2$ | 4C FD E0 7B 09 96 37 9A 99 F0 4B 68 DE 13 0E 59 |
| $H_3$ | 8C 04 5C 44 97 24 48 76 A8 38 64 7C A6 37 B2 45 |
| $G$ | 7E 1A 9A 5E 0E F0 94 7F 25 CB 94 85 26 1C 98 5C |

MAC=7E 1A 9A 5E 0E F0 94 7F

——使用数据比特串 1 和填充方法 3

| | |
|---|---|
| 密钥$K$ | 01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10 |
| 密钥$K'$ | 41 49 D2 AD ED 94 56 68 1E C8 B5 11 D9 E7 EE 04 |
| 密钥$K''$ | B1 B9 22 5D 1D 64 A6 98 EE 38 45 E1 29 17 1E F4 |
| $e_K(D_1)$ | 03 FE 50 C5 56 A3 DB 0F CA AC F5 A7 C1 C5 0C A9 |

| | |
|---|---|
| $H_1$ | 71 59 35 2B EB 73 9D 5B 12 1F 6B EE E3 04 53 C8 |
| $D_2 \oplus H_1$ | 25 31 5C 58 CB 1A EE 7B 66 77 0E CE 97 61 20 BC |
| $H_2$ | 59 B9 0A 21 38 65 1F 29 56 E9 60 8A A8 09 97 8E |
| $D_3 \oplus H_2$ | 79 D4 6F 52 4B 04 78 4C 76 8F 0F F8 88 64 F6 ED |
| $H_3$ | 6F C3 E0 D2 6E D6 3A 49 CA 0E 12 0D FE FE E9 B1 |
| $G$ | 28 A7 0D 6B CC F7 44 22 46 20 58 AB BC 27 F6 AE |

MAC=28 A7 0D 6B CC F7 44 22

——使用数据比特串 2 和填充方法 1

| | |
|---|---|
| 密钥 $K$ | 01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10 |
| 密钥 $K'$ | 41 49 D2 AD ED 94 56 68 1E C8 B5 11 D9 E7 EE 04 |
| 密钥 $K''$ | B1 B9 22 5D 1D 64 A6 98 EE 38 45 E1 29 17 1E F4 |
| $e_K(D_1)$ | 45 FF A9 48 60 5F 52 E8 F4 EF 21 D5 5C D8 F8 0C |
| $H_1$ | BA 52 E9 0F 9A 59 91 F0 C9 85 C4 56 69 9F 8D 27 |
| $D_2 \oplus H_1$ | 9A 3F 8C 7C E9 38 F6 95 E9 85 C4 56 69 9F 8D 27 |
| $H_2$ | C9 A2 2F 02 4B B4 91 09 CA 79 2B C0 DC 36 67 C1 |
| $G$ | AA 9D B3 D9 65 1F 86 2B 6F 18 D6 74 92 13 25 E0 |

MAC=AA 9D B3 D9 65 1F 86 2B

——使用数据比特串 2 和填充方法 2

| | |
|---|---|
| 密钥 $K$ | 01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10 |
| 密钥 $K'$ | 41 49 D2 AD ED 94 56 68 1E C8 B5 11 D9 E7 EE 04 |
| 密钥 $K''$ | B1 B9 22 5D 1D 64 A6 98 EE 38 45 E1 29 17 1E F4 |
| $e_K(D_1)$ | 45 FF A9 48 60 5F 52 E8 F4 EF 21 D5 5C D8 F8 0C |
| $H_1$ | BA 52 E9 0F 9A 59 91 F0 C9 85 C4 56 69 9F 8D 27 |
| $D_2 \oplus H_1$ | 9A 3F 8C 7C E9 38 F6 95 E9 05 C4 56 69 9F 8D 27 |
| $H_2$ | 4B 22 C9 E3 7B 4A 02 3E 94 89 15 CD DE 26 3D 74 |
| $G$ | 94 94 76 D3 5F 17 26 1E 1F B8 C4 39 6D 62 DC 05 |

MAC=94 94 76 D3 5F 17 26 1E

——使用数据比特串 2 和填充方法 3

| | |
|---|---|
| 密钥 $K$ | 01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10 |
| 密钥 $K'$ | 41 49 D2 AD ED 94 56 68 1E C8 B5 11 D9 E7 EE 04 |
| 密钥 $K''$ | B1 B9 22 5D 1D 64 A6 98 EE 38 45 E1 29 17 1E F4 |
| $e_K(D_1)$ | 42 39 BB 2B 9A A0 09 0B E0 D3 48 17 C7 2B 1B C8 |
| $H_1$ | BA 12 90 4E 07 EE D2 CE 34 64 A3 51 3C 4D 6C 95 |
| $D_2 \oplus H_1$ | EE 7A F9 3D 27 87 A1 EE 40 0C C6 71 48 28 1F E1 |
| $H_2$ | BB 96 37 C0 FB F2 B7 86 BC 54 12 27 20 67 26 53 |

| | |
|---|---|
| $D_3 \oplus H_2$ | 9B FB 52 B3 88 93 D0 E3 9C 54 12 27 20 67 26 53 |
| $H_3$ | CC B9 49 0D D1 B4 EA A3 82 0E 8C 5B F5 53 F8 59 |
| $G$ | C9 D3 4E 16 C4 9A B6 43 57 A2 61 8D EB D1 03 2F |

MAC=C9 D3 4E 16 C4 9A B6 43

## A.6 MAC算法5（CMAC）

这里密钥 $K$ = 01 23 45 67 89 AB CD EF FE DC BA 98 76 5 4 32 10（16进制）。掩码密钥 $K_1$ 和 $K_2$ 由 $K$ 使用密钥诱导方法2得到。MAC的比特长度 $m$ 等于64。

——使用数据比特串 1 和填充方法 4

| | |
|---|---|
| 密钥 $K$ | 01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10 |
| $S = e_K(0^{128})$ | 26 77 F4 6B 09 C1 22 CC 97 55 33 10 5B D4 A2 2A |
| $K_1$ | 4C EF E8 D6 13 82 45 99 2E AA 66 20 B7 A9 44 54 |
| $K_2$ | 99 DF D1 AC 27 04 8B 32 5D 54 CC 41 6F 52 88 A8 |
| $H_1$ | 45 FF A9 48 60 5F 52 E8 F4 EF 21 D5 5C D8 F8 0C |
| $D_2 \oplus H_1$ | 65 92 CC 3B 13 3E 35 8D D4 89 4E A7 7C B5 99 6F |
| $G$ | 69 2C 43 71 00 F3 B5 EE 2B 8A BC EF 37 3D 99 0C |

MAC=69 2C 43 71 00 F3 B5 EE

——使用数据比特串 2 和填充方法 4

| | |
|---|---|
| 密钥 $K$ | 01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10 |
| $S = e_K(0^{128})$ | 26 77 F4 6B 09 C1 22 CC 97 55 33 10 5B D4 A2 2A |
| $K_1$ | 4C EF E8 D6 13 82 45 99 2E AA 66 20 B7 A9 44 54 |
| $K_2$ | 99 DF D1 AC 27 04 8B 32 5D 54 CC 41 6F 52 88 A8 |
| $H_1$ | 45 FF A9 48 60 5F 52 E8 F4 EF 21 D5 5C D8 F8 0C |
| $D_2 \oplus H_1$ | 65 92 CC 3B 13 3E 35 8D D4 6F 21 D5 5C D8 F8 0C |
| $G$ | 47 38 A6 C7 60 B2 80 FC 0C 8A 8A F3 88 6E 9F 5D |

MAC=47 38 A6 C7 60 B2 80 FC

## A.7 MAC算法6（LMAC）

这里密钥 $K^*$ = 01 23 45 67 89 AB CD EF FE DC BA 98 76 5 4 32 10（16进制）。密钥 $K$ 和 $K'$ 由 $K^*$ 使用密钥诱导方法1得到。MAC的比特长度 $m$ 等于64。

——使用数据比特串 1 和填充方法 1

| | |
|---|---|
| 密钥 $K^*$ | 01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10 |
| 密钥 $K$ | 4E 59 5B F0 3F 23 BD 10 32 9B AF 56 98 E8 98 EC |
| 密钥 $K'$ | B3 13 6C 04 4E 95 48 2D 4F 65 2E 69 4F 27 41 CD |
| $H_1$ | 7A 0F 91 F3 1D D3 4E F2 23 4B C9 05 EA DC 80 14 |

| | |
|---|---|
| $D_2 \oplus H_1$ | 5A 62 F4 80 6E B2 29 97 03 2D A6 77 CA B1 E1 77 |
| $G$ | B3 8A 96 19 5B AA 61 FC D7 82 05 9F 35 9E 6E D5 |

MAC=B3 8A 96 19 5B AA 61 FC

——使用数据比特串 1 和填充方法 2

| | |
|---|---|
| 密钥 $K^*$ | 01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10 |
| 密钥 $K$ | 4E 59 5B F0 3F 23 BD 10 32 9B AF 56 98 E8 98 EC |
| 密钥 $K'$ | B3 13 6C 04 4E 95 48 2D 4F 65 2E 69 4F 27 41 CD |
| $H_1$ | 7A 0F 91 F3 1D D3 4E F2 23 4B C9 05 EA DC 80 14 |
| $D_2 \oplus H_1$ | 5A 62 F4 80 6E B2 29 97 03 2D A6 77 CA B1 E1 77 |
| $H_2$ | 9C 93 76 0B A0 AF E2 15 51 78 5D 0C 61 3B B3 61 |
| $D_3 \oplus H_2$ | 1C 93 76 0B A0 AF E2 15 51 78 5D 0C 61 3B B3 61 |
| $G$ | A0 C4 65 EE 58 96 97 2F 83 37 AA 1F 92 C9 9D 10 |

MAC=A0 C4 65 EE 58 96 97 2F

——使用数据比特串 1 和填充方法 3

| | |
|---|---|
| 密钥 $K^*$ | 01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10 |
| 密钥 $K$ | 4E 59 5B F0 3F 23 BD 10 32 9B AF 56 98 E8 98 EC |
| 密钥 $K'$ | B3 13 6C 04 4E 95 48 2D 4F 65 2E 69 4F 27 41 CD |
| $H_1$ | 75 9B E0 D4 71 87 52 EB 89 2A 40 E5 99 43 E9 16 |
| $D_2 \oplus H_1$ | 21 F3 89 A7 51 EE 21 CB FD 42 25 C5 ED 26 9A 62 |
| $H_2$ | 45 A0 43 8C E1 42 AC 38 5E 08 33 B9 CF AE 46 33 |
| $D_3 \oplus H_2$ | 65 CD 26 FF 92 23 CB 5D 7E 6E 5C CB EF C3 27 50 |
| $G$ | 43 05 0D 51 C6 56 AE 60 BE 27 3F BE A4 87 0E F1 |

MAC=43 05 0D 51 C6 56 AE 60

——使用数据比特串 2 和填充方法 1

| | |
|---|---|
| 密钥 $K^*$ | 01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10 |
| 密钥 $K$ | 4E 59 5B F0 3F 23 BD 10 32 9B AF 56 98 E8 98 EC |
| 密钥 $K'$ | B3 13 6C 04 4E 95 48 2D 4F 65 2E 69 4F 27 41 CD |
| $H_1$ | 7A 0F 91 F3 1D D3 4E F2 23 4B C9 05 EA DC 80 14 |
| $D_2 \oplus H_1$ | 5A 62 F4 80 6E B2 29 97 03 4B C9 05 EA DC 80 14 |
| $G$ | 8C F6 E6 43 14 FE F4 17 3E 7A 8A EB 67 C5 BE 57 |

MAC=8C F6 E6 43 14 FE F4 17

——使用数据比特串 2 和填充方法 2

| | |
|---|---|
| 密钥 $K^*$ | 01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10 |
| 密钥 $K$ | 4E 59 5B F0 3F 23 BD 10 32 9B AF 56 98 E8 98 EC |
| 密钥 $K'$ | B3 13 6C 04 4E 95 48 2D 4F 65 2E 69 4F 27 41 CD |
| $H_1$ | 7A 0F 91 F3 1D D3 4E F2 23 4B C9 05 EA DC 80 14 |

| | |
|---|---|
| $D_2 \oplus H_1$ | 5A 62 F4 80 6E B2 29 97 03 CB C9 05 EA DC 80 14 |
| $G$ | 60 DD 95 5E D0 CA 3D 7A 64 22 71 74 DD 98 DD 81 |

MAC=60 DD 95 5E D0 CA 3D 7A

——使用数据比特串 2 和填充方法 3

| | |
|---|---|
| 密钥 $K^*$ | 01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10 |
| 密钥 $K$ | 4E 59 5B F0 3F 23 BD 10 32 9B AF 56 98 E8 98 EC |
| 密钥 $K'$ | B3 13 6C 04 4E 95 48 2D 4F 65 2E 69 4F 27 41 CD |
| $H_1$ | C8 70 DD 59 02 49 30 88 3A DD 10 80 CE 76 D7 63 |
| $D_2 \oplus H_1$ | 9C 18 B4 2A 22 20 43 A8 4E B5 75 A0 BA 13 A4 17 |
| $H_2$ | 6E 09 28 0E C2 2C E9 1B ED 84 45 12 FF B5 C0 E6 |
| $D_3 \oplus H_2$ | 4E 64 4D 7D B1 4D 8E 7E CD 84 45 12 FF B5 C0 E6 |
| $G$ | 61 E0 00 49 E2 69 62 A3 6F EDBA 8D 4F 52 F0 AD |

MAC=61 E0 00 49 E2 69 62 A3

## A.8 MAC算法 7（TrCBC）

这里使用的密钥是 $K = 01\ 23\ 45\ 67\ 89\ AB\ CD\ EF\ FE\ DC\ BA\ 98\ 76\ 54\ 32\ 10$ （16进制）。MAC的比特长度 $m$ 等于64。

——使用数据比特串 1 和填充方法 4

| | |
|---|---|
| 密钥 $K$ | 01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10 |
| $H_1$ | 45 FF A9 48 60 5F 52 E8 F4 EF 21 D5 5C D8 F8 0C |
| $D_2 \oplus H_1$ | 65 92 CC 3B 13 3E 35 8D D4 89 4E A7 7C B5 99 6F |
| $G = H_2$ | 16 E0 29 04 EF B7 65 B7 06 45 9C 9E DA BD B5 19 |

MAC=16 E0 29 04 EF B7 65 B7

——使用数据比特串 2 和填充方法 4

| | |
|---|---|
| 密钥 $K$ | 01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10 |
| $H_1$ | 45 FF A9 48 60 5F 52 E8 F4 EF 21 D5 5C D8 F8 0C |
| $D_2 \oplus H_1$ | 65 92 CC 3B 13 3E 35 8D D4 6F 21 D5 5C D8 F8 0C |
| $G = H_2$ | 42 1A D1 69 0A A1 52 E2 84 6F A2 A5 D8 34 45 A9 |

MAC=84 6F A2 A5 D8 34 45 A9

## A.9 MAC算法 8（CBCR）

这里使用的密钥是 $K = 01\ 23\ 45\ 67\ 89\ AB\ CD\ EF\ FE\ DC\ BA\ 98\ 76\ 54\ 32\ 10$ （16进制）。MAC的比特长度 $m$ 等于64。

——使用数据比特串 1 和填充方法 4

| | |
|---|---|
| 密钥 $K$ | 01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10 |
| $H_0 = e_K(0^{128})$ | 26 77 F4 6B 09 C1 22 CC 97 55 33 10 5B D4 A2 2A |

| | |
|---|---|
| $D_1 \oplus H_0$ | 72 1F 9D 18 29 A8 51 EC E3 3D 56 30 2F B1 D1 5E |
| $H_1$ | 18 68 26 95 3B 1F BC 63 F1 84 06 C0 D9 42 4E 52 |
| $D_2 \oplus H_1$ | 38 05 43 E6 48 7E DB 06 D1 E2 69 B2 F9 2F 2F 31 |
| $G = H_2$ | E4 0E D7 9C 31 49 A1 C9 D4 2F 04 C4 23 04 99 35 |

MAC=E4 0E D7 9C 31 49 A1 C9

——使用数据比特串 2 和填充方法 4

| | |
|---|---|
| 密钥 $K$ | 01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10 |
| $H_0 = e_K(0^{128})$ | 26 77 F4 6B 09 C1 22 CC 97 55 33 10 5B D4 A2 2A |
| $D_1 \oplus H_0$ | 72 1F 9D 18 29 A8 51 EC E3 3D 56 30 2F B1 D1 5E |
| $H_1$ | 18 68 26 95 3B 1F BC 63 F1 84 06 C0 D9 42 4E 52 |
| $D_2 \oplus H_1$ | 38 05 43 E6 48 7E DB 06 D1 04 06 C0 D9 42 4E 52 |
| $G = H_2$ | A9 9D 13 01 3E 89 2E E2 C2 5B E2 DA AA 6C 82 E8 |

MAC=A9 9D 13 01 3E 89 2E E2

# 附 录 B
（资料性附录）
MAC 算法的安全性分析

本附录讨论了本部分中MAC算法的安全强度。它的目标是协助本部分的使用者选择合适的MAC算法。

假定分组密码的密钥长度为 $k$ 比特，MAC算法的密钥长度为 $k^*$ 比特，所以 $k^* = k$ 或 $k^* = 2k$ 。

本附录中， $\mathrm{MAC}_K(D)$ 表示用密钥为 $K$ 的MAC算法对消息 $D$ 进行计算所得到的MAC。

为了确定MAC算法的安全强度，本附录考虑了如下两类攻击：

——**伪造攻击**：此类攻击是在没有密钥 $K$ 的情况下，对消息 $D$ 预测 $\mathrm{MAC}_K(D)$ 。如果攻击者能够对一个消息成功预测其 MAC，那么称他有能力“伪造”。实际攻击经常要求伪造是可验证的，也就是说，以接近 1 的概率确认伪造的 MAC 是正确的。在许多应用中消息有特定的格式，这就意味着对消息 $D$ 有额外限制。

——**密钥恢复攻击**：此类攻击根据大量的（消息，MAC）对找到 MAC 算法的密钥 $K$ 。密钥恢复攻击比伪造攻击更强大，因为它一旦成功就可进行任意伪造。

一个攻击的可行性依赖于攻击者已知和选择的（消息，MAC）对数目以及离线加密的次数。

对MAC算法可能的攻击描述如下，但是这里并不保证列举了所有的攻击。前两种攻击是一般性的，它们对任何MAC算法都有效。第三种适用于迭代的MAC算法。随后的三种攻击只对本标准中的一个或多个特定MAC算法有效（更多信息请参阅[16,21,22,26,27,28]）。

——**猜测 MAC**：这种伪造是不可验证的，成功概率为 $\max(1/2^m, 1/2^{k^*})$ 。这种攻击适用于所有的 MAC 算法，只有合适地选择 $m$ 和 $k^*$ 才能够抵抗这种攻击。

——**密钥穷搜索**：这种攻击需要运行平均 $2^{k^*-1}$ 次 MAC 算法，并且需要 $k^*/m$ 对（消息，MAC）以唯一确定密钥。同样这种攻击适用于所有 MAC 算法，合适地选择 $k^*$ 能够抵抗这种攻击。另外，MAC 算法使用者也可阻止攻击者获得 $k^*/m$ 对（消息，MAC）以抵抗这种攻击。例如，如果 $k^* = 64$ 且 $m = 32$ ，给定的对（消息，MAC）相当于 $2^{32}$ 个密钥。如果每次使用 MAC 算法后都改变密钥，那么密钥穷搜索攻击并不比猜测 MAC 攻击更有效。

——**生日攻击**[26,28]：如果攻击者获得接近 $2^{n/2}$ 对（消息，MAC），将有很高的概率找到消息 $D$ 和 $D'$ ，使得： $\mathrm{MAC}_K(D) = \mathrm{MAC}_K(D')$ ，并且 $H_a$ 的值在两次 MAC 计算中是相等的（被称为内部碰撞）。如果消息 $D$ 和 $D'$ 构成内部碰撞，那么对任意的比特串 $Y$ 都有 $\mathrm{MAC}_K(D \| Y) = \mathrm{MAC}_K(D' \| Y)$ 。这就构成了一种伪造，当攻击者得到比特串 $D \| Y$ 的 MAC 时，就能够预测比特串 $D' \| Y$ 的 MAC。同样，这种伪造依赖于消息的特殊格式，可能对许多应用没有威胁。但是，这种攻击的扩展版本在消息格式方面有更大的灵活性。这种攻击需要一个比特串、大约 $2^{n/2}$ 对已知（消息，MAC）和 $\min\{2^{n-m}, 2^{n/2}\}$ 对选择（消息，MAC）。

使用填充方法 3 和在要处理的消息前面加上一个序列号分组并不能避免生日攻击（具体参见文献[17]）。

——**简单伪造**：若采用填充方法 1，那么攻击者可轻易地增加或删除消息最后的几个“0”比特，却保持 MAC 不变。这就意味着填充方法 1 只能用在 MAC 算法使用者事先知道消息长度的情况下，或者消息最后有不同个数的“0”却意义相同的情况。

——**异或伪造**：若 MAC 算法 1 采用填充方法 1 或 2，并且 $m = n$ ，那么就可能存在一个简单的异或伪造攻击。简单来讲，假如消息 $D$ 或其被填充后的数据 $\overline{D}$ 只有一个分组长度（如果使用填充方法 2，假定 $D$ 的长度小于 $n$ 比特）。$v$ 表示将比特串最右侧的比特“1”以及随后的所有“0”

去除的操作（假如$\overline{v(X)}$表示$v(X)$经填充方法2得到的数据，则$\overline{v(X)}=X$）。

假定攻击者获得了$\mathrm{MAC}_K(D)$。如果采用填充方法1，可知$\mathrm{MAC}_K(\overline{D}\|(\overline{D}\oplus\mathrm{MAC}_K(D)))=\mathrm{MAC}_K(D)$。类似地，如果使用填充方法2，可知$\mathrm{MAC}_K(\overline{D}\|v(\overline{D}\oplus\mathrm{MAC}_K(D)))=\mathrm{MAC}_K(D)$。这就意味着攻击者可构造一个伪造。

注意到：即便MAC算法的密钥仅使用一次，这类攻击仍然适用。如果攻击者获得了$\mathrm{MAC}_K(D)$和$\mathrm{MAC}_K(D')$。如果采用填充方法1，经过类似的推导可知：$\mathrm{MAC}_K(\overline{D}\|(\overline{D'}\oplus\mathrm{MAC}_K(D)))=\mathrm{MAC}_K(D')$（这里$D$的长度任意，而$D'$的长度必须是一个分组）。类似地，如果采用填充方法2，可知$\mathrm{MAC}_K(\overline{D}\|v(\overline{D'}\oplus\mathrm{MAC}_K(D)))=\mathrm{MAC}_K(D')$（这里$D$的长度任意，而$D'$的长度必须是一个分组）。

此外，对于填充方法1，如果攻击者知道$\mathrm{MAC}_K(D)$，$\mathrm{MAC}_K(D\|Y)$和$\mathrm{MAC}_K(D')$，那么就知道了$\mathrm{MAC}_K(D'\|Y')=\mathrm{MAC}_K(D\|Y)$，其中$Y'=Y\oplus\mathrm{MAC}_K(D)\oplus\mathrm{MAC}_K(D')$（这里假定$D$和$Y$的比特长度为$n$的整数倍）。这也构成了一个伪造，因为攻击者在获得了两个已知消息和一个选择消息所对应的MAC之后，能够对比特串$D'\|Y'$伪造MAC。对于填充方法2，类似（但较为复杂）的伪造攻击也有效。

值得注意的是，上述伪造依赖于消息的特殊格式，可能对许多应用没有威胁。

采用填充方法3可抵抗这种攻击。

若$m<n$，这种攻击仍然适用，但是更加困难，需要额外的$2^{(n-m)/2}$对选择（消息，MAC）[21]。这种攻击对使用两个相同密钥（$K'=K$）的MAC算法2也适用，不过这里要求$Y$包含至少两个分组，并且其前面$n$比特为“0”。

——**捷径密钥恢复**：基于内部碰撞的密钥恢复攻击适用于某些MAC算法。比如MAC算法3[22,24,27])，以及采用填充方法1、2[19]或3[18]的MAC算法4。部分密钥恢复攻击适用于MAC算法5[24]。获得部分密钥后，容易构造伪造。

如下的表格比较了本部分各MAC算法的安全强度。这里假定底层分组密码算法没有任何弱点。表B.1列举了各MAC算法的主要性质。因为采用填充方法1存在简单伪造攻击，所以MAC算法1、2、3、4和6只采用填充方法2和3。表B.2和表B.3针对采用$n=64$和$k=56$（比如DES算法[8]）分组密码的MAC算法，介绍了最好的攻击。表B.4和表B.5针对采用$n=64$和$k=128$分组密码的MAC算法，表B.6和表B.7针对采用$n=128$和$k=128$分组密码的MAC算法。在这些情形中，无需将MAC算法的密钥长度加倍，因此仅考虑MAC算法1和2。其中的大部分攻击源自[16,17,18,19,20,21,22,26,27,28]。攻击复杂度使用4元组$[\alpha,\beta,\gamma,\delta]$描述，其中$\alpha$表示离线加密的次数，$\beta$表示已知（消息，MAC）对的数目，$\gamma$表示选择（消息，MAC）的数目，$\delta$表示在线验证的次数。

在假设分组密码是伪随机置换的情况下，文献[16]对输入长度固定且为分组长度整数倍的MAC算法1做了分析，给出了安全强度的一个下界，证明了其安全性；文献[25]对输入长度是分组长度任意整数倍的MAC算法2做了分析，同样证明了其安全性。文献[16,25]同时说明了在假定底层分组密码没有弱点的条件下，前面所述的很多生日攻击接近于最好的攻击。

**表B.1　MAC算法的特性；密钥个数表示相互独立的分组密码密钥个数，效率表示处理长度为$tn$比特的比特串所用的加密次数**

| 序号 | MAC算法 | 初始变换 | 最终变换 | 输出变换 | 截断操作 | 填充 | 密钥个数 | 效率 |
|---|---|---|---|---|---|---|---|---|
| 1.1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | $t+1$ |
| 1.2 | 1 | 1 | 1 | 1 | 1 | 3 | 1 | $t+1$ |
| 2.1 | 2 | 1 | 1 | 2 | 1 | 2 | 1 | $t+2$ |
| 2.2 | 2 | 1 | 1 | 2 | 1 | 2 | 2 | $t+2$ |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 3 | 3 | 1 | 1 | 3 | 1 | 2 | 2 | $t+3$ |
| 4.1 | 4 | 2 | 1 | 2 | 1 | 2 | 2 | $t+3$ |
| 4.2 | 4 | 2 | 1 | 2 | 1 | 3 | 2 | $t+3$ |
| 5 | 5 | 1 | 3 | 1 | 1 | 4 | 1[a] | $t$[a] |
| 6.1 | 6 | 1 | 2 | 1 | 1 | 2 | 1 | $t+1$ |
| 6.2 | 6 | 1 | 2 | 1 | 1 | 2 | 2 | $t+1$ |
| 7 | 7 | 1 | 1 | 1 | 2 | 4 | 1 | $t$ |
| 8 | 8 | 3 | 4 | 1 | 1 | 4 | 1 | $t+1$ |
| [a] MAC 算法 5 需要预计算一次加密，存储额外的两个 $n$ 比特密钥。 | | | | | | | | |

表 B.2　当 $n=64$、$k=56$ 和 $m=64$ 时的安全强度估计；安全强度由四个数字表示：离线加密的次数，已知（消息，MAC）的数目，选择（消息，MAC）的数目以及在线验证的次数

| | 密钥恢复 | | 伪造 | | |
|---|---|---|---|---|---|
| 序号 | 密钥穷搜索 | 捷径密钥恢复 | 猜测 MAC 值 | 异或 | 生日伪造 |
| 1.1 | $[2^{56},1,0,0]$ | —— | $[0,0,0,2^{56}]$ | $[0,1,0,0]$ | $[0,2^{32},1,0]$[a]<br>$[0,1,2^{32},0]$ |
| 1.2 | $[2^{56},1,0,0]$ | —— | $[0,0,0,2^{56}]$ | —— | $[0,2^{32},1,0]$[a]<br>$[0,1,2^{32},0]$ |
| 2.1 | $[2^{56},1,0,0]$ | —— | $[0,0,0,2^{56}]$ | —— | $[0,2^{32},1,0]$[a]<br>$[0,1,2^{32},0]$ |
| 2.2 | $[2^{112},2,0,0]$ | $[2^{57},2,0,0]$ | $[0,0,0,2^{64}]$ | —— | $[0,2^{32},1,0]$[a]<br>$[0,1,2^{32},0]$ |
| 3 | $[2^{112},2,0,0]$ | $[2^{57},2^{32},0,0]$<br>$[2^{56},1,0,2^{56}]$<br>$[2^{57},2,0,2^{63}]$ | $[0,0,0,2^{64}]$<br>$[0,1,0,2^{56}]$ | —— | $[0,2^{32},1,0]$[a]<br>$[0,1,2^{32},0]$ |
| 4.1 | $[2^{112},2,0,0]$ | $[2^{58},2^{32},2,0]$[a]<br>$[2^{58},1,1,2^{56}]$[a] | $[0,0,0,2^{64}]$<br>$[0,1,0,2^{56}]$[a] | —— | $[0,2^{32},1,0]$[a] |
| 4.2 | $[2^{112},2,0,0]$ | $[2^{58},2^{33},2^{50},0]$[a] | $[0,0,0,2^{64}]$ | —— | $[0,2^{32},1,0]$[a]<br>$[0,0,1,2^{64}]$[a] |
| 5 | $[2^{56},1,0,0]$ | $[0,2^{33},0,0]$[b] | $[0,0,0,2^{64}]$ | —— | $[0,2^{33},0,0]$ |
| 6.1 | $[2^{56},1,0,0]$ | —— | $[0,0,0,2^{56}]$ | —— | $[0,2^{32},1,0]$[a]<br>$[0,1,2^{32},0]$ |
| 6.2 | $[2^{112},2,0,0]$ | $[2^{57},2,0,0]$ | $[0,0,0,2^{64}]$ | —— | $[0,2^{32},1,0]$[a]<br>$[0,1,2^{32},0]$ |
| 8 | $[2^{56},1,0,0]$ | —— | $[0,0,0,2^{64}]$ | —— | $[0,2^{33},0,0]$ |
| [a] 表示采用填充方式 3 并且在数据比特串头部附加一个序列号消息块可避免相应攻击。 | | | | | |
| [b] 仅恢复用于简单伪造的掩码密钥。 | | | | | |

表 B.3　当 $n=64$、$k=56$ 和 $m=32$ 时的安全强度估计；安全强度由四个数字表示：离线加密的次数，已知（消息，MAC）的数目，选择（消息，MAC）的数目以及在线验证的次数

| | 密钥恢复 | | 伪造 | | |
|---|---|---|---|---|---|
| 序号 | 密钥穷搜索 | 捷径密钥恢复 | 猜测 MAC 值 | 异或 | 生日伪造 |
| 1.1 | $[2^{56},2,0,0]$ | —— | $[0,0,0,2^{32}]$ | $[0,2,2^{16},0]$ | $[0,2^{32},2^{32},0]$ [a] |
| 1.2 | $[2^{56},2,0,0]$ | —— | $[0,0,0,2^{32}]$ | —— | $[0,2^{32},2^{32},0]$ [a] |
| 2.1 | $[2^{56},2,0,0]$ | —— | $[0,0,0,2^{32}]$ | —— | $[0,2^{32},2^{32},0]$ [a] |
| 2.2 | $[2^{112},4,0,0]$ | $[2^{57},2^{32},2^{32},0]$<br>$[2^{88},4,0,0]$ | $[0,0,0,2^{32}]$ | —— | $[0,2^{32},2^{32},0]$ [a] |
| 3 | $[2^{112},4,0,0]$ | $[2^{57},2^{32},2^{32},0]$ [a]<br>$[2^{89},2^{32},0,0]$<br>$[2^{57},0,0,2^{48}]$ | $[0,0,0,2^{32}]$ | —— | $[0,2^{32},2^{32},0]$ [a] |
| 4.1 | $[2^{112},4,0,0]$ | $[2^{78},2^{32},2^{50},0]$ [a] | $[0,0,0,2^{32}]$ | —— | $[0,2^{32},2^{32},0]$ [a] |
| 4.2 | $[2^{112},4,0,0]$ | $[2^{78},2^{32},2^{50},0]$ [a]<br>$[2^{64},0,2^{63},2^{57}]$ [a] | $[0,0,0,2^{32}]$ | —— | $[0,2^{32},2^{32},0]$ [a] |
| 5 | $[2^{56},2,0,0]$ | $[0,2^{33},2^{33},0]$ [b] | $[0,0,0,2^{64}]$ | —— | $[0,2^{32},2^{32},0]$ |
| 6.1 | $[2^{56},2,0,0]$ | —— | $[0,0,0,2^{32}]$ | —— | $[0,2^{32},2^{32},0]$ [a] |
| 6.2 | $[2^{112},4,0,0]$ | $[2^{57},2^{32},2^{32},0]$<br>$[2^{88},4,0,0]$ | $[0,0,0,2^{32}]$ | —— | $[0,2^{32},2^{32},0]$ [a] |
| 8 | $[2^{56},2,0,0]$ | —— | $[0,0,0,2^{64}]$ | —— | $[0,2^{32},2^{32},0]$ |
| [a] 表示采用填充方式 3 并且在数据比特串头部附加一个序列号消息块可避免相应攻击。 | | | | | |
| [b] 仅恢复用于简单伪造的掩码密钥。 | | | | | |

表 B.4　当 $n=64$、$k=128$ 和 $m=64$ 时的安全强度估计；安全强度由四个数字表示：离线加密的次数，已知（消息，MAC）的数目，选择（消息，MAC）的数目以及在线验证的次数

| | 密钥恢复 | | 伪造 | | |
|---|---|---|---|---|---|
| 序号 | 密钥穷搜索 | 捷径密钥恢复 | 猜测 MAC 值 | 异或 | 生日伪造 |
| 1.1 | $[2^{128},2,0,0]$ | —— | $[0,0,0,2^{64}]$ | $[0,1,0,0]$ | $[0,2^{32},1,0]$ [a]<br>$[0,1,2^{32},0]$ |
| 1.2 | $[2^{128},2,0,0]$ | —— | $[0,0,0,2^{64}]$ | —— | $[0,2^{32},1,0]$ [a]<br>$[0,1,2^{32},0]$ |
| 2.1 | $[2^{128},2,0,0]$ | —— | $[0,0,0,2^{64}]$ | —— | $[0,2^{32},1,0]$ [a]<br>$[0,1,2^{32},0]$ |
| 5 | $[2^{128},2,0,0]$ | $[0,2^{33},0,0]$ [b] | $[0,0,0,2^{64}]$ | —— | $[0,2^{33},0,0]$ |
| 6.1 | $[2^{128},2,0,0]$ | —— | $[0,0,0,2^{64}]$ | —— | $[0,2^{32},1,0]$ [a]<br>$[0,1,2^{32},0]$ |
| 8 | $[2^{128},2,0,0]$ | —— | $[0,0,0,2^{64}]$ | —— | $[0,2^{33},0,0]$ |
| [a] 表示采用填充方式 3 并且在数据比特串头部附加一个序列号消息块可避免相应攻击。 | | | | | |
| [b] 仅恢复用于简单伪造的掩码密钥。 | | | | | |

表 B.5　当 $n=64$、$k=128$ 和 $m=32$ 时的安全强度估计；安全强度由四个数字表示：离线加密的次数，已知（消息，MAC）的数目，选择（消息，MAC）的数目以及在线验证的次数

| | 密钥恢复 | | 伪造 | | |
|---|---|---|---|---|---|
| 序号 | 密钥穷搜索 | 捷径密钥恢复 | 猜测 MAC 值 | 异或 | 生日伪造 |
| 1.1 | $[2^{128},4,0,0]$ | —— | $[0,0,0,2^{32}]$ | $[0,2,2^{16},0]$ | $[0,2^{32},2^{32},0]$ [a] |
| 1.2 | $[2^{128},4,0,0]$ | —— | $[0,0,0,2^{32}]$ | —— | $[0,2^{32},2^{32},0]$ [a] |
| 2.1 | $[2^{128},4,0,0]$ | —— | $[0,0,0,2^{32}]$ | —— | $[0,2^{32},2^{32},0]$ [a] |
| 5 | $[2^{128},4,0,0]$ | $[0,2^{33},2^{33},0]$ [b] | $[0,0,0,2^{32}]$ | —— | $[0,2^{33},2^{33},0]$ |
| 6.1 | $[2^{128},4,0,0]$ | —— | $[0,0,0,2^{32}]$ | —— | $[0,2^{32},2^{32},0]$ [a] |
| 8 | $[2^{128},4,0,0]$ | —— | $[0,0,0,2^{32}]$ | —— | $[0,2^{33},2^{33},0]$ |
| [a] 表示采用填充方式 3 并且在数据比特串头部附加一个序列号消息块可避免相应攻击。 | | | | | |
| [b] 仅恢复用于简单伪造的掩码密钥。 | | | | | |

表 B.6　当 $n=128$、$k=128$ 和 $m=64$ 时的安全强度估计；安全强度由四个数字表示：离线加密的次数，已知（消息，MAC）的数目，选择（消息，MAC）的数目以及在线验证的次数

| | 密钥恢复 | | 伪造 | | |
|---|---|---|---|---|---|
| 序号 | 密钥穷搜索 | 捷径密钥恢复 | 猜测 MAC 值 | 异或 | 生日伪造 |
| 1.1 | $[2^{128},2,0,0]$ | —— | $[0,0,0,2^{64}]$ | $[0,2,2^{32},0]$ | $[0,2^{64},2^{64},0]$ [a] |
| 1.2 | $[2^{128},2,0,0]$ | —— | $[0,0,0,2^{64}]$ | —— | $[0,2^{64},2^{64},0]$ [a] |
| 2.1 | $[2^{128},2,0,0]$ | —— | $[0,0,0,2^{64}]$ | —— | $[0,2^{64},2^{64},0]$ [a] |
| 5 | $[2^{128},2,0,0]$ | $[0,2^{65},2^{65},0]$ [b] | $[0,0,0,2^{64}]$ | —— | $[0,2^{65},2^{65},0]$ |
| 6.1 | $[2^{128},2,0,0]$ | —— | $[0,0,0,2^{64}]$ | —— | $[0,2^{64},2^{64},0]$ [a] |
| 8 | $[2^{128},2,0,0]$ | —— | $[0,0,0,2^{64}]$ | —— | $[0,2^{65},2^{65},0]$ |
| [a] 表示采用填充方式 3 并且在数据比特串头部附加一个序列号消息块可避免相应攻击。 | | | | | |
| [b] 仅恢复用于简单伪造的掩码密钥。 | | | | | |

表 B.7　当 $n=128$、$k=128$ 和 $m=32$ 时的安全强度估计；安全强度由四个数字表示：离线加密的次数，已知（消息，MAC）的数目，选择（消息，MAC）的数目以及在线验证的次数

| | 密钥恢复 | | 伪造 | | |
|---|---|---|---|---|---|
| 序号 | 密钥穷搜索 | 捷径密钥恢复 | 猜测 MAC 值 | 异或 | 生日伪造 |
| 1.1 | $[2^{128},4,0,0]$ | —— | $[0,0,0,2^{32}]$ | $[0,2,2^{48},0]$ | $[0,2^{64},2^{64},0]$ [a] |
| 1.2 | $[2^{128},4,0,0]$ | —— | $[0,0,0,2^{32}]$ | —— | $[0,2^{64},2^{64},0]$ [a] |
| 2.1 | $[2^{128},4,0,0]$ | —— | $[0,0,0,2^{32}]$ | —— | $[0,2^{64},2^{64},0]$ [a] |
| 5 | $[2^{128},4,0,0]$ | $[0,2^{65},2^{97},0]$ [b] | $[0,0,0,2^{32}]$ | —— | $[0,2^{65},2^{97},0]$ |
| 6.1 | $[2^{128},4,0,0]$ | —— | $[0,0,0,2^{32}]$ | —— | $[0,2^{64},2^{64},0]$ [a] |
| 7 | $[2^{128},4,0,0]$ | —— | $[0,0,0,2^{32}]$ | —— | $[0,2^{65},2^{97},0]$ |

| | | | | | |
|---|---|---|---|---|---|
| 8 | $[2^{128},4,0,0]$ | —— | $[0,0,0,2^{32}]$ | —— | $[0,2^{65},2^{97},0]$ |
| [a] 表示采用填充方式 3 并且在数据比特串头部附加一个序列号消息块可避免相应攻击。 | | | | | |
| [b] 仅恢复用于简单伪造的掩码密钥。 | | | | | |

# 参 考 文 献

[1] ISO 8731-1:1987, Banking — Approved algorithms for message authentication — Part 1: DEA
[2] ISO 8732:1988, Banking — Key management (wholesale)
[3] ISO/IEC 8825-1:2002, Information technology — ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)
[4] ISO 9807:1991, Banking and related financial services — Requirements for message authentication (retail)
[5] ISO/IEC 10116:2006, Information technology — Security techniques — Modes of operation for an n-bit block cipher
[6] ISO/IEC 11770 (all parts), Information technology — Security techniques — Key management
[7] ISO 11568 (all parts), Banking — Key management (retail)
[8] ANSI X3.92:1981, Data Encryption Algorithm
[9] ANSI X9.9:1986, Financial Institution Message Authentication (Wholesale)
[10] ANSI X9.19:1986, Financial Institution Retail Message Authentication
[11] ANSI X9.24-1:2004, Retail Financial Services Symmetric Key Management — Part 1: Using Symmetric Techniques
[12] NIST Special Publication 800-38B: 2005, Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, May 2005
[13] GB/T 9387.2-1995, 《信息处理系统 开放系统互联 基本参考模型 第2 部分：安全体系结构》
[14] GB/T 15843.1-2017, 《信息技术 安全技术 实体鉴别 第1 部分：概述》
[15] GB/T 17901.1-1999, 《信息技术 安全技术 密钥管理 第1部分：框架》
[16] M. Bellare, J. Kilian, and P. Rogaway, ‘The security of cipher block chaining’, Advances in Cryptology, Proceedings Crypto'94, LNCS 839, Y. Desmedt, Ed., Springer-Verlag, 1994, pp. 341-358
[17] K. Brincat and C.J. Mitchell, ‘New CBC-MAC forgery attacks’, Information Security and Privacy, ACISP 2001, LNCS 2119, V. Varadharajan and Y. Mu, Eds., Springer-Verlag, 2001, pp. 3-14
[18] D. Coppersmith, L.R. Knudsen, and C.J. Mitchell, ‘Key recovery and forgery attacks on the MacDES MAC algorithm’, Advances in Cryptology, Proceedings Crypto 2000, LNCS 1880, M. Bellare, Ed., Springer-Verlag, 2000, pp. 184-196
[19] D. Coppersmith and C.J. Mitchell, ‘Attacks on MacDES MAC algorithm’, Electronics Letters, Vol. 35, No. 19, 1999, pp. 1626-1627
[20] T. Iwata and K. Kurosawa, ‘OMAC: One-key CBC MAC’, Proceedings Fast Software Encryption 2003, LNCS 2887, T. Johansson, Ed., Springer-Verlag, 2003, pp. 129-153
[21] L. Knudsen, ‘Chosen-text attack on CBC-MAC’, Electronics Letters, Vol. 33, No. 1, 1997, pp. 48-49
[22] L. Knudsen and B. Preneel, ‘MacDES: MAC algorithm based on DES’, Electronics Letters, Vol. 34, No. 9, 1998, pp. 871-873
[23] C.J. Mitchell, ‘Key recovery attack on ANSI retail MAC’, Electronics Letters, Vol. 39, 2003, pp. 361-362
[24] C.J. Mitchell, ‘Partial key recovery attack on XCBC, TMAC and OMAC’, Cryptography and Coding: Proceedings 10th IMA International Conference, LNCS 3796, N. Smart, Ed., Springer-Verlag, 2005, pp. 155-167 (See also: Royal Holloway, University of London, Mathematics Department Technical Report

RHUL-MA-2003-4, August 2003, 15 pages)

[25] E. Petrank and C. Rackoff, ‘CBC MAC for real-time data sources’, Journal of Cryptology, Vol. 13, No. 3, 2000, pp. 315-338

[26] B. Preneel and P.C. van Oorschot, ‘MDx-MAC and building fast MACs from hash functions’, Advances in Cryptology, Proceedings Crypto'95, LNCS 963, D. Coppersmith, Ed., Springer-Verlag, 1995, pp. 1-14

[27] B. Preneel and P.C. van Oorschot, ‘A key recovery attack on the ANSI X9.19 retail MAC’, Electronics Letters, Vol. 32, No. 17, 1996, pp. 1568-1569

[28] B. Preneel and P.C. van Oorschot, ‘On the security of iterated Message Authentication Codes’, IEEE Transactions on Information Theory, Vol. 45, No. 1, January 1999, pp. 188-199

[29] L. Zhang, W. Wu, P. Wang, and B. Liang, ‘TrCBC: Another look at CBC-MAC’, Information Processing Letters, Vol. 112, No. 7, 2012, pp. 302-307

[30] L. Zhang, W. Wu, L. Zhang, and P. Wang, ‘CBCR: CBC MAC with rotating transformations’, Science China Information Sciences, Vol. 54, No. 11, 2011, pp. 2247-2255